ISSN 0951-9335
02
9 770951 933023

SINCLAIR QL WORLD

Volume 3 issue 2 price £2.50

SUPER GOLD CARD
Miracle System releases Gold Card Upgrade

DIY TOOLKIT
SCSI routines for the QL

REMODELLING THE QL
Hard disk interface and tower casing from Jurgen Falkenberg

DIGITAL PRECISION

THE POWER BEHIND THE BUTTON

1984-1994

THE BEST SOFTWARE FOR THE BEST COMPUTER

A PRIDE IN ITS LOGIC

NO IDEALISTIC PRIG

SPIRITED LOGICIAN

DIGITAL PRECISION

A LOGIC IN ITS PRIDE

PERFECTION SPECIAL EDITION

POWER

PERFECTION SPECIAL EDITION has 253 (two hundred and fifty three) direct/menu commands (not counting options in sub-menus), plus 32 special characters (like Bold on) that can be inserted 'directly' plus intelligent (and now excellently documented) macros. Comparisons with other word processors on the subject of power are hence quite unnecessary.

EASE OF USE

Independent reports, customer feedback and published reviews (of its less able but still excellent predecessor, PERFECTION) leave one in no doubt as to which word processor is friendliest – PERFECTION SPECIAL EDITION, with its intuitive, silky handling. Uniquely, it has two operating modes, with both menus (visible or invisible – they even look like Quill's) and direct commands (for when you familiarise yourself with the system). Uniquely, both modes are 're-entrant' (so you can use any menu option or direct command while you are in the middle of performing another option or command – block handling, etc, becomes a dream). Uniquely, PERFECTION SE has fully automatic memory management, grabbing and releasing RAM instantly as your document grows or shrinks – programs without this don't take full advantage of the multi-tasking abilities of the QL! Uniquely, PERFECTION SE leaves you in the driving seat, not juggling things around 'underfoot' while you are typing. Uniquely, PERFECTION SE allows up to nine different documents to be handled simultaneously from one copy of the program – each with totally independent margin, tab, justification, control panel, etc, settings. Uniquely, each document can itself have up to six environment settings, each settable or recallable instantly with a single keypress combination. Each document can have any number (up to 500,000 on GOLD CARD) of candidate blocks! Each document can have two independent windows (of any depth, of any (but same) width across) 'on to' it, even with overlapping text – that allows you to edit in one place while viewing another, to compare 'before editing' with 'after editing' (you can arrange to have one window remain 'frozen' in time), etc. Uniquely, we realise how much faster it is to type in something like CTRL/SHIFT/F5 than (say) F3 F3 R – both involve three keys, but as the former doesn't require the keys to be pressed in just one specific order, or to be released in any order at all (together will do), it is in practice twice as fast as the latter, where no key may be pressed until its predecessor is released. PERFECTION SE takes advantage of all this – it is the little things that count! Uniquely, by providing eight user-definable strips, PERFECTION SE allows you to cope with printers of the future, not just the printers that now exist – you can attach the strips to any printer features. Uniquely, PERFECTION SE's status lines give full information on all relevant global settings. And the manual has an index. Also, it has all the important bits at the front.

WYSIWYG?

By the latest definition of this term, neither is PERFECTION SE fully WYSIWYG, nor are other QL word processors. WYSIWYG means what you see on screen is exactly what you get on paper. Exactly – down to every wiggle in every character in every font.

To get true WYSIWYG, use PERFECTION SE's fully automatic link (supplied as part of PERFECTION SE) to PROFESSIONAL PUBLISHER, where you will get 100% WYSIWYG. 100%? Yes, 100%. With this combination, adjust the horizontal and vertical magnification on your monitor (ie calibrate it once and for all so screen circles correspond to same-diameter printed circles – poor monitors may distort a little bit at the edges). Now you can place your printed output from PERFECTION via PUBLISHER over your monitor screen, and get a match that is more perfect than is your eyesight. Now that is WYSIWYG.

SUPERB PRINT QUALITY & FLEXIBILITY

Uniquely, using the aforementioned automatic link, you can output PERFECTION SE documents using over a thousand fonts (a huge variety of styles and sizes, supplied on the PUBLISHER and TOOLBOX disks) on virtually any printer – from the humblest Epson RX80, Brother M1009 or Star LC10 (which are all single font machines when used with most word processors) to top-end lasers. *You are not limited to the fonts built into the printer!!* All PERFECTION SE **bold**/underlined/ *italics*/^{super}/$_{sub}$, etc, settings are preserved. Proportional spacing and micro-justification are automatic, even when you mix fonts of differing widths *and heights* (even on the same line), vary line spacings, etc. Uniquely, you are not trapped with one type of micro-justification (ie adding all the space between words, and using the predefined widths of characters as their separation) – with our system, you can vary (in 5% steps) the proportion of micro-spaces added between words to that added between characters (the latter in proportion to their *individual* widths). Settings around 65%-35% – not the 100%-0% forced upon you by some other word processors – seem to give the most pleasing results. Uniquely, you are not limited to mere rectangular columns plus headers/footers – that's all the rest can do – you can output in any sequence to any number of frames (text flowing from one to the next), each of any shape – irregular polygons of up to 66 sides, circles, multi-column or part-column boxes (hundreds of types of borders, thousands of textures), doughnuts, wrap-around shapes, even re-entrant ones ('join-the-dots' type borders, even with intersecting edges) – all with micro-justification and proportional spacing! Look at the example on this page. Of course, if super fancy output or special effects are not of the essence, PERFECTION SE's direct printer output is more than capable of meeting your needs.

THE FASTEST

For benchmarking, we've used an unimpeachable file – not one created specially – a public domain version of the first book of The King James Bible, all fifty chapters of the book of Genesis. This came to **one hundred and forty pages**, well over **forty two thousand words** excluding headers and footers, well over **two hundred and twelve thousand characters** excluding justification ones and **one thousand five hundred and thirty three indexed verses**!! We didn't use a smaller file (as used to benchmark other programs) as PERFECTION SE's timings for most operations then become impossible to stopwatch (too fast!). The hardware used for all timings was GOLD CARD: speeds would be **further improved by over three times** using the SUPER GC. Of course, LIGHTNING SE was used. File operations were to ramdisk: normal slave blocks would give identical times. All settings on **everything** were for maximum speed, except where indicated to the contrary – we have the sense **not** to force full speed upon you in operations like scrolling and global Search & Replace. PERFECTION SE's speed for these is switchable (at run-time and when configuring), as too great a speed may cause overshoot (with scrolling) or fatal alteration (if there is human error inputting the target or replace strings). Here are the benchmarks for this huge file:

Load 140 pages: 0.6 seconds (yes 0.6, not 6!) ☆ Import 140 pages: 0.6 seconds (yes 0.6, not 6!) ☆ Save 140 pages: 0.5 seconds (yes 0.5, not 5!) ☆ Export 140 pages: 0.5 seconds (yes 0.5, not 5!) ☆ Case-sensitive search from top for word at bottom: 0.4 seconds (yes 0.4, not 4!) ☆ The same, but case case-insensitive: 0.5 seconds (yes 0.5, not 5!) ☆ Case-sensitive search backwards from bottom for word at top: 0.4 seconds (yes 0.4, not 4!) ☆ The same, but case-insensitive: 0.5 seconds (yes 0.5, not 5!) ☆ Automatic Search & Replace, in Fast (No Query) mode, of last 600 occurrences: 7.4 seconds (same length replace string); 7.7 seconds (shorter replace string); 10.5 seconds (longer replace string – longer time as we deliberately chose a high *density* of replaces to handicap PERFECTION SE into auto-managing memory – without causing any heap fragmentation, but still with only a 0.005 second overhead per replace!) ☆ Automatic Search & Replace in Slow ('Querying') mode: arbitrarily slow, typically 30 times slower – because we deliberately allow for human response time (in case you want to abort) before proceeding from one replace to the next – booby prize to anyone for benchmarking us on this setting!! ☆ Scrolling 100 lines of text, up or down, by full-width screen page: 1.5 seconds ☆ Scrolling 100 lines of text on full-width screen, line by line, in slow (full) mode: 5.7 seconds (down)/5.8 seconds (up) ☆ As above, but in medium speed mode: 4 seconds ☆ The same, but in fast mode and default settings: 13.5 seconds to scroll through the whole massive document, averaging 0.23 seconds per 100 pages (!) – and this could be made up to ten times faster by reconfiguring PERFECTION SE ☆ Reformatting paragraphs, changing margins, justification, etc, of existing text: c5 times faster than predecessor ☆ Inserting (or undoing) emphasised, underlined, italics, superscript, subscript, 8 strips, 6 environment settings: Instant (i.e. immeasurable) ☆ Navigation to line or page or to top or bottom or to 8 markers or to highlights/blocks: Instant ☆ Setting new margins, justification, etc: Instant ☆ Deleting block of 100 pages: 0.3 (yes, 0.3 not 3!) seconds ☆ Copying/moving block of 100 pages (not just 10!), downwards or upwards: 3.4 seconds (yes, including all the time for automatic memory management and anti-fragmentation – other programs are light-years behind) ☆ Spellcheck as you type: Ten times faster than anyone can possibly type ☆ Spellcheck all 140 pages in the document using the 350,000 word Mega Dictionary: 3.9 seconds (20 'errors' – like 'pluckt'!) ☆ And using our tiny dictionary (well, tiny by our standards – large by comparison with most others): 5.1 seconds (566 'errors') ☆ Time taken to create user dictionary from the results of the second spellcheck (566 errors): 0.8 seconds to extract all 'errors' from document and clean document; 1.9 seconds to create a full user dictionary therefrom and also a sorted, duplicate-free wordlist file (for browsing) ☆ Spellcheck file (ASCII or native): Even faster. ☆ Print first 10 pages to file: 3.5 seconds. ☆ Change every occurrence of God to ***God*** in bold underlined italics, strip 2 – 9.5 seconds!

TECHNICAL NOTES Reformatting is the amendment of a section of previously-entered text to conform to margin, indentation, justification and pagination settings after the user returns to it and makes alterations, either by hand (by over-typing, deleting, adding or otherwise changing) or using search and replace, merge etc. PERFECTION SE lets the user pre-configure, or tune at run-time, the desired reformatting behaviour. The options are to either select Never (most suitable for technical users, and what all previous PERFECTIONs did: you had to initiate the reformat of the re-edited para), Instant (= 0.1 seconds, giving in-situ real-time automatic reformatting as-you-type: common in word processors, and irritating to the eye) or User-delay, the most flexible setting (giving slightly delayed auto-updating of lower text). On User-delay the user is free to set any delay from 0.2 seconds to 99.9 seconds in 0.1 second steps. About 1.5 seconds is best for sedate typists and 0.3 seconds for speed demons. This means that you are not hassled by continuing screen changes on lines below the one you are editing and concentrating upon, or shufflings around on the current line caused by right or centre justification. When you pause in your typing for longer than the set delay, PERFECTION SE automatically tidies up, without you having to do anything. On User-delay, if you navigate or progress off the line, or invoke any menu or command (including Save, Print etc.), an auto-reformat occurs instantly. This means that you are never left with the document in the wrong state. With these options, you have the best of all possible worlds.

Also, SHIFT/CAPS now obeys the indent margin (which matters if the cursor is on the first line of a para) and leaves the cursor position unaltered within the text. Other reformatting commands are unaltered, so you can still step through paras reformatting manually as you go, if you wish. The maximum number of lines, characters, words, lines, pages etc have all been increased (effectively to infinity: e.g., the new limit on characters is 30 million-million, this up from 2 million, restrictive in Super Gold Card / QXL days!). Also, the new version (starting with v5.13) is even faster, and its handling of complex search/replaces (say, involving end of line codes) has been optimised. PERFECTION SE really is superb!

DIGITAL PRECISION PRICES

	£
3-D PRECISION CAD SYSTEM	49.95 d
ADVENTURE CREATION TOOL SPECIAL EDITION	49.95 r
ARCADIA GAME	9.95 s
ARCHIVE DEVELOPMENT SYS + RUN-TIME MODULE	29.95 a
ARCHIVE TUTORIAL	19.95 a
BETTER BASIC EXPERT SYSTEM	24.95 a
BLOCKLANDS GAME	9.95 s
CASH TRADER v3.3 + ANALYSER	99.95 c
COMPARE FILE UTILITY	19.95 a
COPY UTILITY FOR BACKING UP	9.95 a
CPORT BASIC TO C AUTO-TRANSLATOR WITH C68 COMPILER	99.95 c
CPORT BASIC TO C AUTO-TRANSLATOR	89.95 c
DATABASE ANALYSER FOR ARCHIVE	19.95 a
DAT-APPOINT APPOINTMENT DATABASE SYSTEM	19.95 d
DESKJET/LASERJET DRIVER FOR PRO PUBLISHER	19.95 c
DESKTOP PUBLISHER SPECIAL EDITION	39.95 c
DESKTOP PUBLISHER	24.95 d
DIGITAL C COMPILER	29.95 a
DIGITAL C SPECIAL EDITION	49.95 a
DISKTOOL WITH QUICKDISK	19.95 b
DROIDZONE GAME	9.95 a
DR-DOS v7 STATE OF THE ART PC OPERATING SYSTEM	89.95 r
EDITOR SPECIAL EDITION	49.95 d
EDITOR	29.95 a
EYE-Q GRAPHICS SYSTEM	39.95 a
FONT ENLARGER	19.95 r
GAMES COMPENDIUM (ALL FIVE GAMES)	29.95 a
GRAFIX 24-PIN UNIVERSAL PRINTER DRIVER	14.95 r
HARDBACK + FINDER HARD DISK MANAGER	49.95 c
IDIS INTELLIGENT DISASSEMBLER	24.95 a
IDIS SPECIAL EDITION	39.95 d
LIGHTNING ACCELERATOR	24.95 a
LIGHTNING SPECIAL EDITION ACCELERATOR	49.95 h
LIGHTNING GOLD SPECIAL EDITION	39.95 a
MAILMERGE DATABASE SYSTEM	19.95 a
MEDIA MANAGER SPECIAL EDITION	49.95 d
MEDIA MANAGER	24.95 a
MEGA DICTIONARY	29.95 g
MICROBRIDGE	39.95 d
NAMES AND ADDRESSES DATABASE SYSTEM	19.95 d
PAYROLL SYSTEM	49.95 c
PC CONQUEROR GOLD SPECIAL EDITION WITH DR-DOS v7	179.95 g
PC CONQUEROR GOLD SPECIAL EDITION	99.95 g
PC CONQUEROR WITH DR-DOS v7	139.95 r
PC CONQUEROR	59.95 r
PEDIT PRINTER DRIVER FOR XCHANGE	19.95 a
PERFECTION PLUS SPECIAL EDITION WITH SPELLCHECKER	139.95 c
PERFECTION PLUS WITH SPELLCHECKER	99.95 r
PERFECTION SPECIAL EDITION WORD PROCESSOR v5	99.95 c
PERFECTION SPECIAL EDITION DEMO VERSION + SAMPLE DATA	19.95 c
PERFECTION WORD PROCESSOR v3	59.95 d
PERFECT POINTER TOOLS	29.95 a
PROFESSIONAL ASTROLOGER WITH ASTRONOMER	69.95 a
PROFESSIONAL ASTROLOGER	59.95 a
PROFESSIONAL ASTRONOMER	29.95 s
PROFESSIONAL PUBLISHER TOOLBOXES (BOTH PARTS)	49.95 r
PROFESSIONAL PUBLISHER TOOLBOX PART ONE	29.95 r
PROFESSIONAL PUBLISHER TOOLBOX PART TWO	29.95 r
PROFESSIONAL PUBLISHER	89.95 c
QFLICK CARD INDEX SYSTEM	29.95 a
QKICK FRONT END SYSTEM	24.95 a
QMATHS MATHEMATICAL SYSTEM PART ONE	69.95 c
QMATHS MATHEMATICAL SYSTEM PART TWO	59.95 c
QMATHS MATHEMATICAL SYSTEM (BOTH PARTS)	99.95 c
QMON MACHINE CODE MONITOR	39.95 a
QUICKLASER PUBLISHER DRIVER FOR LASERJET/DESKJET PRINTERS	19.95 c
RECOVER ARCHIVE UTILITY	19.95 a
REVERSI GAME	9.95 s
SEDIT + SCREENPRINT ARCHIVE UTILITIES	19.95 a
SOLUTION PC EMULATOR WITH DR-DOS v7	109.95 r
SOLUTION PC EMULATOR	29.95 r
SPELLCHECKER WITH DICTIONARIES	49.95 d
SUCCESS CP/M EMULATOR	49.95 b
SUPERBASIC MONITOR	24.95 a
SUPERFORTH COMPILER WITH REVERSI	39.95 a
SUPER ASTROLOGER	24.95 s
SUPER BACKGAMMON GAME	9.95 s
SUPER SPRITE GENERATOR	29.95 a
TOOLKIT III WITH ROM	49.95 h
TOOLKIT III	29.95 a
TRANSFER UTILITY SPECIAL EDITION	29.95 b
XREF SUPERBASIC PROGRAM ANALYSER	29.95 a
TRANSFER UTILITY	9.95 b
TURBO BASIC COMPILER + TOOLKIT	79.95 a
ULTRAPRINT SCREEN DUMP UTILITY	19.95 a
XREF SUPERBASIC PROGRAM ANALYSER	29.95 a

KEY a=Available either on cartridge or disk; b=Disk only; c=Minimum 512K exp: disk only; d=Minimum 256K exp: either cartridge or disk; r=Minimum 256K exp: disk only; s=Cartridge only; g=Minimum 1.5Mb RAM: disk only disk; h=ROM + (cartridge or disk)

AND NOW - BEST-EVER SPECIAL DEALS...

5% off 2 programs
10% off 3 programs
20% off 4 programs
30% off 5 programs
40% off 6+ programs!

SUPER GOLD CARD

This amazing product is the way forward for the QL. Like Gold Card before it, the brand-new **Super Gold Card** is a plug-in disk interface and RAM expansion that works on all QL versions. Incredibly, it is *over three times* speedier than Gold Card, with *over twice* the RAM and with many enhancements. It complements LIGHTNING SPECIAL EDITION like nothing else, squeezes the best out of TURBO (which was designed with 32-bit CPUs in mind) and really accelerates PC CONQUEROR. **Super Gold Card** is actually as fast, or slightly faster, than the much vaunted QXL: also, it is **100%** QL-compatible **now**, and no PC is needed. The table below really says it all:

System → *↓ Features*	*Bare* *QL*	*TRUMP* *CARD I*	*GOLD* *CARD*	**SUPER** **GOLD CARD**
Relative Speed	1x	1.8x	7x	**25x !**
Motorola CPU	68008	68008	68000	**M68020**
Clock Frequency	7.5MHz	7.5MHz	16MHz	**24MHz**
Bus width	8 bit	8 bit	16 bit	**32 bit**
RAM fitted	128Kb	896Kb	1,920Kb	**3,968Kb**
RAM access speed	Slow	OK	Fast	**Twice as fast**
PCB population	V.high	High	V.low	**V.low**
Physical dimensions	Monolith	Full-size	Half-size	**Half-size**
Lock-up frequency	Ouch!	Occasional	V.rare	**Won't**
Battery Backup Clock	No	No	Yes	**Yes**
Clock Protection level	N/A	N/A	Modest	**High**
Toolkit II + Manual	No	Yes (early vns)	Yes	**Enlarged**
Sub-directory support	No	No	Yes	**Yes**
Parallel/Centronics port	No	No	No	**Yes**
Spooler/Screendump/Ramdisks	No	Yes	Yes	**Yes**
Speedup switch (Screen#2)	No	No	No	**Yes**
Future hi-res graphics	No	No	No	**Planned**
Disk drives supported	N/A	SD/DD	SD/DD/HD/ED	**SD/DD/HD/ED**
Max no: of disk drives	0	2	3	**4**
Max sectors/disk	N/A	1,440	6,400	**6,400**
Max disk transfer rate	N/A	30Kb/sec	120Kb/sec	**>120Kb/sec**
Peripheral card tolerance	OK	No	No	**OK**
SCSI-2 compatibility to-be	No	?	No	**Yes**
Miracle/DP Warranty	No	No	2 years	**2 years**
DIY/Kit incorporability	Yes	No	No	**Yes**
Overall Rating by DP	2%	10%	30%	**110%**

And to the Very Best news: from DP, SUPER GOLD comes SUPER CHEAP!
SUPER GOLD CARD, plus a no-limit extra 20% SOFTWARE DP DISCOUNT VOUCHER, plus a FREE mystery DP program, plus a FREE Dust Cover, will cost you a mere £375 ✓✓✓✓
Less £125 if part-exchanging your standard 2Mb Gold Card
Add £125 for ED 6400-sector Disk Drive (PSU, cased, cables).

OTHER HARDWARE EXCHANGED AND SOLD BY ARRANGEMENT. PLEASE ORDER NOW: WE EXPECT VERY HIGH DEMAND FOR SUPER GOLD CARDS, AND WE DON'T WISH TO DISAPPOINT. INTERNATIONAL RAM PRICES ARE UNSTABLE AND PRICE HIKES MAY BE INEVITABLE. CONSEQUENTLY, THE ABOVE COMBINATION OFFER COULD BE WITHDRAWN WITHOUT NOTICE...

The software discount must be taken at the same time as the main order, and CAN be combined with the SPECIAL DEALS discounts. For example, if you chose six DP programs of total list price £100, you would only have to pay £100 -40% -20% = £48 for them! And, of course, you would also get the two gifts absolutely free, and a 4Mb SUPER GOLD CARD too!

*TERMS & CONDITIONS * We accept payment by VISA, Mastercard, Access, Eurocard, Eurocheque, £ cheque drawn on a UK branch of any bank or building society, postal order, cash, travellers cheques or direct funds transfer to our bank account #60327808 at Barclays Bank plc, South Chingford Branch (sort code 20-53-00), NE London Business Centre, PO Box 2403, London N18 2BY. If you must use another currency, add 7.5%. If you must use another form of cheque, add £15. * Upgrades cost only the difference in current price plus £10; return the original disk and manual. But do NOT return the manual if the upgrade is to a "SPECIAL EDITION" of your original program: you will be sent useful supplementary documentation. Upgrades from LIGHTNING are the exceptions to the last rule! * Only DP programs and upgrades (i.e. not DR-DOS) count towards discounts. * Prices are all-inclusive for UK mail-order. Rest of Europe, add 5%. Elsewhere, add 10% (Air Mail). * We aim for 24 hour turnaround. * DP & Digital Precision are trading names of Digital Precision Ltd, Company No.1833989, registered in England & Wales. * Orders using your own paper are also welcome!*

✂✂✂✂✂✂✂✂✂✂✂✂✂✂✂✂✂✂✂✂✂✂✂✂

Post to: *DIGITAL PRECISION*, 222 The Avenue, London E4 9SE, UK
or phone on 081-527 5493 from UK, +44 81-527 5493 rest of the world

Name:__________ Address:__________

Products ordered: __________

on mdv/3.5"disk/5.25"disk* (no: of sectors on disk-format = 360/720/1440/2880/6400*; if 5.25"/720, specify 40/80* track) *=*delete as needed.*

Signature: __________ Amount: £__________

Card No: __________ __________ Expiry date:__________

QL Contents

VOL. 3 ISSUE 2

Coming soon
On the cards: World Map, moving trom Quill to another WP, particularly Perfection; Biorhythms; Biorhythms.

Editor
Helen Armstrong

Publisher
Mark Kasprowicz

Advertising Manager
Jim Peskett

Creative Director
John Stanley

Graphic Artist
Stevie Billington

Magazine Services
Linda Miller, Frances Maxwell, Pauline Wakeling, Val Brandon

Sinclair QL World, Published by Arcwind Ltd. The Blue Barn, Tew Lane, Wootton, Woodstock, Oxon. OX7 1HA
Tel: 0993 811181
Fax: 0993 811481
ISSN 026806X

If you have any comments or difficulties please write to the editor and we will do our best to deal with your problem in the magazine, though we cannot guarantee individual replies.
Back issues are available from the publisher price £2.50 UK, £2.99 Europe. Overseas rates on request.

Subscriptions from: Arcwind The Blue Barn, Tew Lane, Wootton, Woodstock, Oxon. OX7 1HA
UK: £23.40
Europe: £32.90
Rest of World: £40.90

Reprographic Services: Eclipse, Brook Street, Watlington, Oxon. OX9 5JH.
Distributed by: Seymour Press Ltd., Windsor House, 1270 London Road, Norbury, London, SW16 4DH

QL Scene

Version 2.00 for LineDesign

Progs of Belgium have issued version two of their **graphics** program LineDesign. Drawing on their experiences since the well-received first issue of LineDesign in 1993, Version 2 is "much faster, more user friendly, supports more printers, can handle text better, is supplied with more fonts (about 130), can include bitmaps ... too many changes to name them all!" say Progs. Current users who want the upgrade, send the appropriate payment with your program disk only - keep your clipart disk. The upgrade will comprise a new program disk, a new set of fonts, and a new improved manual.

LineDesign still needs around 1 megabyte of free memory to work comfortably. Progs are therefore offering a special deal to users who order LineDesign or DataDesign along with a Gold Card, QXL (Miracle) or QVME (Jochen Merz) card. Contact Progs (Joachim and Nathan Van Auwera, tel/fax 010 32 16 48 8952, UK to Belgium) for further details.

Prices for LineDesign are (in Belgian francs, inclusive of tax and postage): LineDesign V2 BeF 5000 (4350 outside EC vat area); LineDesign upgrade BeF 1000 (930) until 1 April 1994 (after which it will be BeF 2500); LD with QVME or Gold Card: please phone; LD with 2MB QXM BeF 22500 (19,000); LD with 5MB QXL BeF 27,000 (22,800); LD with 8MB QXL BeF 31,500 (26,500). DataDesign 3.08 with engine 3.09 BeF 3000 (2700); DD API BeF 1000 (930); DD with Gold Card or QVME please call; DD with QXL - prices as for LineDesign, but subtract BeF 1800 (1500) in each case.

Another new product from Progs is **ProForma**, a programmers' library which gives C programmers (users of C68) the ability to write programs which use the Progs Font and Raster Manager. This is the dedicated graphics library which was also used for the development of LineDesign 2.00. The library supports scalable outlines fonts and vector graphics, and can also be used from Assembler. ProForma costs BeF 5000.

Progs accept payment in Eurocheques (Belgian francs), Visa and Eurocard/Mastercard (give card number and expiry date), and transfer to their postal account (Call for details). Customers living outside the European Community pay the bracketed prices exempt from VAT..

Like a number of other QL dealers, Progs have assigned the handling of some of their older popular products to another dealer to concentrate on their more recent developments. Anyone who wants to buy or find out about The Painter, The ClipArt, or Qractal should contact Dilwyn Jones at DJC. (See back cover of QL World.)

Progs, Haachtstraat 92, 3020 Veltem, Belgium.

Dilwyn Jones Software is now doing Creative Codeworks' old program Speedscreen, with some improvements. Ask Dilwyn for information. Dilwyn is also hoping to visit the IQLR/Rhode Island show in the United States on 14th May 1994.

Plus4 in Danish

Software87 and Danish suppliers Interlogic have collaborated to produce a Danish version of Text87 Plus4. The Danish version is complete with Danish demo disk, Danish Spellchecking library and an 120-page Danish manual for DKK 949. The demo disk is free on receipt of a 720K formatted disk and sufficient return postage. All this and other Software 87 products can be obtained via **Interlogic, Gunther Strube, Gl. Kongevej 37, 2th, DK-1610 Kopenhagen V, Denmark..**

Don't forget the Quanta Bristol Show at the Walton Park Hotel, Avon, near Bristol, on 27th March

Albin Hessler's desktop program Cueshell has just appeared in Beta Test form, and we are waiting for the general release as soon as the manual is finished.

Page Designer Update

Dilwyn Jones has issued updates to two recent programs, **Page Designer 3** and **Convert-PCX**.

PD3 1.07 now has a Zoom pixel-editing facility, allowing pixel work to be done at times-five magnification; red and green paint fill as well as standard white; a new printer-driver two-pass overstrike option, and the ability of print a secreen-sized section of the page - especially helpful for exporting a screen from PD3 to another program.

Convert-PCX 1.02 has various bug-fixes, including one which prevented PCX pictures converted to PC3 from merging correctly with PC3 pages, and and a problem with conversions where line-buffer lengths had not been specified, previously resulting in some files crashing Convert-PCX.

For new users, Convert-PCX costs £10, and PD3 costs £40, or £25 to upgrade from PD2. Upgrades for current owners of PC3 or Convert-PCX are free on return of the appropriate master disk plus suitable return postage (2 International Reply Coupons outside the UK). PC3 owners should send disk 1 of their four disks only.

Dilwyn Jones Computing, 41 Bro Emrys, Tal-Y-Bont, Bangor, Gwynedd, LL57 3YT. Tel. 0248 354023.

QL Scene

SUPER GOLD CARD NOW HERE

Miracle Systems are taking orders for **new-version Gold Card**, now named the Super Gold Card. The new Card, based on a 68020 running at 24 megahertz, has a "lot in common" with the original Gold Card, but will give its users about three times the speed of the Old Gold, and twice the memory with 4 MHz of ram.

The Super Gold Card also has a **Centronics board** to allow graphics to print five to 10 times faster than previously, in line with demanding up to date QL programs like LineDesign and the newest Text87 which found Serial printing too slow. There are also two disk drive ports, incorporating the functionality of Miracle's Disk Adaptor as well.

The Super Gold Card retails at £375 fully inclusive (£325 outside the EU VAT area), or £225 (£200) to upgrade from a present old Gold Card. Miracle has stopped making the old Gold Card, and will rely on trade-in cards to supply reconditioned **second-user Gold Cards** for about £150. But Miracle's Stuart Honeyball says "Quite a lot of people have decided they want a new one, not an upgrade. The are putting the old one on their second QLs."

This is however good news for less hardware-rich users, giving them an opportunity to get into the Gold Card at a much reduced price on their first (or only!) machines.

Other features include a "virtually crashproof clock. We've never had any trouble with the clock going on the Gold Card", said Stuart, "But some people say that some software crashes the clock. We've also condescended to people who put their QLs into those big cases and can't find a 9V supply! Instead of having to rip the rectifier out, they'll find the new Gold Card has a power socket so that you can run 5 volts straight into it." And there are other "small things" which Miracle have tweaked.

The Super Gold Card has already made its appearance in public at Edinburgh and other shows in February and March. "It actually runs all the programs we've tried," said Stuart, "and the **compatibility** seems to be very good. If anyone wants to try their software on any of our hardware, tell them to bring it on a disk to one of our demos, and we'll run it! We haven't found anything that doesn't run yet, and if there are things, we want to know about them." In fact, says Stuart, if any QL group anywhere in the UK or Europe wants a demonstration, contact Miracle and they'll fix something up.

Miracle will be in the USA at Rhode Island on May 14th with all their kit.

Says Stuart: "We've taken everything we've learned in the lifetime of the Gold Card and put it right on the Super Gold Card. Mostly they're nothing to do with the Gold Card itself, but we feel we have to get round them anyway."

Miracle will also be **exhibiting at the Bristol Workshop, Walton Park Hotel, Avon, near Bristol on 27th March, and the Quanta AGM at Park Hotel, Tynemouth, Newcastle upon Tyne on 23/24th April.**

All Formats Diary

In 1994, the All Formats Computer Fairs will be held monthly at three northern venues, and other places.

The **monthly fairs** will take place at the National Motocycle Museum, junction 6 of the M42, opposite the National Exhibition Centre entrance, Haydock Park Racecourse, J23 of the M6, between Manchester and Liverpool, and the Washington Leisure Centres at the top of the A1(M). A smaller number of fairs will be in Bristol, Glasgow, the South East, Belfast and Cardiff. In the next two months:

Mar 6 Brunel Centre, Templemeads Station, Bristol Mar 13 Surbiton Tolworth Recreation Centre, A3 19 Haydock Park Racecourse, J23 M6 Mar 20 National Motorcycle Museum, near NEC, J6 M42 Mar 26 Washington Leisure Centre, Distric 1 Mar 27 Glasgow Woodside Hall, St George's Cross Apr 9 Brentwood Centre, off A12, J28 M25 Apr 10 Cardiff Univeristy Union, Park Place Apr 16 Haydock Park Apr 17 Belfast Ulster Hall, Bedford St. Apr 23 Washington Leisure Centre Apr 24 National Motorcycle Museum.

Attendance at All Formats Fairs by QL specialists is now very occasional. Fairs are open from 10am to 4pm, entrance adult £4 (£3 with voucher), £2 child; £2 all comers after 2pm. A supply of "£1 off" vouchers can be obtained by sending an SAE to **All Formats Fairs, Maple Leaf, Stretton-on-Fosse, Moreton-in-Marsh, Gloucestershire GL56 9QX. Stands at the fairs cost £60 tel. 0608 662212.**

QL Press

Recently out, with a write-up on Perfection, columns on C and Public Domain, and games reviews among other features, is **QReview**, editor Bruce Nicholls' follow-up to the now-deceased *QL Games Review* and *QL Technical Review* from CGH. Vol 1 issue II was scheduled for November 1993 but is running a little late - not alone! Bruce also has back issues of QLTR, QLLR, and *QL Adventurers' Forum.* Price list from **Quo Vadis Design, 57 Shaftesbury Road, Romford, Essex RM1 2QL. Tel. 0708 755759**

Sinclair QL User Club eV's magazine, Quasar 35 (December 1993) has a general news column, a write-up on an HDD interface by Dirk Steinkopf, and mention of MasterBasic, fractals (with listings), the QL in Austria, and some reviews and opinion columns. A4, in German. Contact SQL User Club eV via **Franz Herrmann, Talstrasse 21, d-W5460 Ochenfels, West Germany.**

OPEN CHANNEL

Open Channel is where you have the opportunity to voice your opinions in Sinclair QL World. Whether you want to ask for help with a technical problem, provide somebody with an answer, or just sound off about something which bothers you, write to: Open Channel, QL World, The Blue Barn, Tew Lane, Wootton, Woodstock OX7 1HA.

Dual Response

I am able to make some comments on two letters recently published in Open Channel. The first is from J Roy Goodall of Belize City (II.8) concerning the transfer of data from a Psion Organiser Model XP. He and I corresponded about this problem a couple of years ago, but could not find a solution. I have since managed to overcome the problems I was having, by reducing the Baud Rate on both machines to 1200 and not set it at 9600 as suggested in the Transform manual.

However, for the modest investment of £25, A **Hermes** chip allows the Baud rate to be set to 9600 on both machines giving fast and reliable data transfer between machines. It's great to have something that works as it should.

The second letter was from Roy Wood in Germany in II.10, concerning inter alia the Glossary function of Quill in Xchange. I have been able to get it to work but not as I would like. Firstly, unlike the glossary of Turbo Quill, each assigned key and its keypresses are not saved to disk until Quill is exited. It is therefore necessary to exit Quill by the Quit command in order for the Glossary to be saved. Furthermore, it seems that glossary files must all be kept on the same disk. If the default device is changed to flp2_, a glossary file on the Xchange disk (flp1_) will load but not be saved if any changes/additions are made. Simon Goodwin did allude to both these points in his assessment of Xchange in issue II.7 of QL World.

I do have a request for help of my own, however. If I use an NLQ printer driver for my Citizen 120-D, which performed faultlessly in Quill, why does the Xchange-dat version insist on adding about 30 blank lines between each new page? This does not happen with the other drivers I have set for draft copy, rough copy or Abacus printouts (condensed print).

Finally, I have been encouraged by some members of the Leeds/Bradford Quanta subgroup to report a use for **Minerva**'s second screen. I have adapted the boot file for Pete Etheridge's Bridge program, which was available through Microdrive Exchange, to set up the second screen to list the Play and Hand options, which have to be entered by number (0 to 24). I could never remember the meanings of the numbers or letters, and being lazy couldn't be bothered to have the manual alongside the monitor. The meanings of the numbers are now available at the touch of Ctrl-Tab.

The listing follows (first the Boot file):

```
10 MODE 4
15 MODE 96,-1: Rem :
Switch to second screen
20 OPEN #3,con_512x256
a0x0:PAPER #3,0:INK #3,4
30 PRINT #3,,:UNDER
#3,1:PRINT #3;"<<<<<< THE
HAND OPTIONS
>>>>>>":UNDER#3,0
:
Lines 40 - 230 contain
"PRINT #3" statements such
as above
:
240 CLOSE #3
250 MODE 96,-1 : Rem :
Switch back to 'amin' screen
260 LRUN FLP1BRBOOT:
Rem: This is original Boot
file
```

The original boot file (renamed brboot) has had to be modified also:

```
130 rompos=PEEK L(PEEK
L(163960)+4)+42
```

(which was part of a DEF PROC to redefine five characters as symbols for the card suits and 10's) has become:

```
130 rompos=PEEK_L(PEEK_
L(196728)+4)+42
```

because the second screen being activated has moved the beginning of the ram from $28000 to $36000 - see page 207 of Jan Jones' *Definitive QL Handbook*.

If anyone else has found a practical use for Minerva's second screen I would be very interested to hear of it as I feel it is an underused facility of an excellent rom.

Robert Goodall
Halifax

Cryptic Mouse

No problem letting you know what I like and what I don't like - I just like the QL World, even when Bryan Davies gets a bit philosophical, and I don't like it when Simon Goodwin gets cryptical as in his second article on Sermouse (II.11), where he explains Listing Two as generating code for the "three button Mouse System PC serial mouse handler for SER2". Does he just mean "to operate a three-button mouse", or what?

Although a little late I also want to express my gratitude to Robin Stevenson who obviously took my remark seriously enough to publish an update (II.5) on his fascinating Desktop program. I feel quite honoured.

Last of all I wish you and your staff happy days and a prosperous new year.

Albert van Rheenen
Amsterdam

Simon Goodwin says: "Mouse Systems is the name of the company which makes various different mice, including a three button model. The project needs a three-button mouse. PC Serial defines the type of mouse - not a PC bus mouse of the motherboard type - the "handler" is software dealing directly with the hardware, and SER2 is for the second QL serial port where mice are usually plugged. All the terms define a requirement for the mouse.

"If Albert is getting a bad line message, the computer is not recognising his command. The most likely answer is that either he hasn't LOADed and LINKed the mouse software, or else he has a JM or AH rom, and is trying to use commands before LOADing and LINKing and typing NEW between linking and using. In which case, load the mouse driver, type NEW and try again."

Get in Touch!

As for the **New User Guide**, well really, correct errors in the existing manual by all means, but it only covers Minerva, Turbo Toolkit and **TK2** commands. What about TK3 by DP, System by Merz or Megatoolkit by DJC, to mention but three? I realise that there is little new software becoming available, and if it was not for people like DP, Progs, Software 87 and Merz I think we would have seen the demise not only of QL World but of the QL as well.

What do we want? Well, more of the **Troubleshooter** type of report, more user help, more comparative reviews, news about the future (and not to be advised about events that have already taken place). What about answering people's questions? Someone a month or two ago asked how to make the input of characters idiot-proof. He, poor soul, is still waiting. Any number of QL users would be more than able to help him out, I would for one.

A G Kendall
Kings Langley
Herts

The New User Guide aimed to cover the most widely-quoted systems, which were, as you say, TK2, Turbo Toolkit, and Minerva. Today these are still the most widely used. If we were starting now we would plan a section on the Pointer Environment as well, but this was covered last year by Ian Bruntlett, at great length!

Well, how can readers help each other if they don't get in touch? All you have to do is offer and we will put you in touch! We don't print everything in Open Channel, some things are too complicated. Someone has already sent an answer to the query you mention. Don't hesitate - if you have an answer, which will help someone, get in touch.

Beware Laps!

I am writing in response to Michael Jonas's letter in II.10. This is a warning to anyone using their QL as a **lapwarmer**! When I first bought my QL in 1985, I was living in bedsit accommodation, and did not have room for a table, or the luxury of a monitor. I used to sit on my bed with my QL on my lap, hooked up to the TV (no memory expansion or disk drives, either ... was it really so long ago?). As Michael says, the QL is an excellent lapwarmer, with the hottest part over your right knee. However, after several months of usage, the QL began to lose its colour display once warned up. THe length of time before the display turned to monochrome got shorter and shorter, until, in the end, there was **no colour at all**.

As the time, I had very little experience with computers and how they work, so I had no idea how to fix my trusty QL. Then, one month, an issue of *QL User* (anyone remember it?) contained an article which solved my problem. Using the QL as a lapwarmer had overheated the crystal that controls the colour display. The crystal had a value of around 6.433 MHz (I can't remember the exact value) and had also been used in the ZX Spectrum. As a friend of mine had a scrap board, he gave me the crystal off that. Taking a soldering iron to my QL, the replacement chip was soon installed, and with it a full colour display. After this, I used a book to insulate the QL from my lap so that it would not overheat.

Beware! Using your QL as a lapwarmer may damage its health!

Ian Turner
Albrighton
Wolverhampton

Does anyone have any statistics on the relative heat-dispersal gradients of different book formats as compared to the human knee?

Concept Errors

A number of inaccuracies crept into the **New User Guide** in *QL World* Vol. II.12 and I would be grateful for some space to put them right. The standard QL's MC68008 CPU actually works at 7.5MHz, and its memory chips are rated at 150 microseconds with a refresh cycle of 250 Hz, rather than the figures quoted in the article. My only excuses for these errors are a failing memory and the inability to find my QL technical reference notes at a critical time.

The article's claim that the MC68000 instruction set has "hundreds" of commands could be misleading. Stuart Honeyball has pointed out that there are in fact 56 main instructions for machine code programmers to learn, with each one having several variations. The MOVE instruction alone can be expressed in 240 different ways, covering different addressing modes and byte, word and long word options. As one of the joys of MC68000 programming is the orderliness of its command language, I am happy to clarify the *New User Guide* by saying that the MC68000 instruction set is 56 items long with hundreds of variations!

Mike Lloyd
Gloucestershire

Editor's Notebook

The Monitor from Hell has behaved perfectly this month, but what has happened to our hold-over page files? Just another one of life's little mysteries. Fortunately (for a wonder) I have everyone's words tucked away on a personal backup. Really we should have everything on floppy disk too, but there's always a time lag between working and housekeeping when things get busy. Be warned!

The show season is starting in earnest, with QL meetings in Bristol, Newcastle (the Quanta AGM) and Rhode Island, USA, in the next couple of months.

No, *Club Access* isn't one of the Lost Files - I just haven't managed to squeeze it in. A new address for Qliper: Salvador Merino, Ctra Cadiz, Ceramicas Mary, 29640 Terreblanca del Sol, Spain.

Please note - we don't have any backissues before Arcwind took over. Try Qubbesoft, SJPD or other general QL dealers.

TROUBLESHOOTER!

Bryan Davies looks at the latest versions of Line Design and SMS2.

Some QL areas are still very active, and none more so than the one in **Belgium** where **LineDesign** is being developed. The version sent to me early in January '94 was marked "V2.00B" ("B" for beta test), and there have no doubt been further changes since then. This version is a very different beast from the initial one; there are big improvements in both speed and facilities. All of a sudden we are getting much closer to the kind of graphical functions that have been available to users of some other computers for some time, but which have until now - remained only wishes for QL users.

Here is some of the recent information from **Progs**, the originators of LineDesign: there are now 130 founts; special characters, character kerning, bitmaps, thick lines, pie shapes, and squares with rounded corners, are available. **PROforma**, the Progs Font and Raster Manager, is a dedicated graphics library which supports scalable founts and vector graphics. It has been used to develop LineDesign, and is now available separately for C programmers.

Software87 are understood to be offering an update on the **Text87 Publisher's Pack**, to work with version 2 of LineDesign.

The earlier generation of Progs software - The Painter, The ClipART and Qractal - are now handled by Dilwyn Jones Computing.

Speed Thrills

One of the claims that Digital Precision have always made for **Perfection** is that it is a faster word-processing program than both the QL competition and the big names on the PC. There was quite a lot of discussion some time ago on whether or not Perfection was actually faster than Text87. Plenty of room for discussion there, but the argument that Perfection is faster than WP programs on the PC is clearer in some ways. For instance, WP programs running under the Windows GUI are mostly distinctly slower than their DOS-based forerunners, and they get very much slower when large documents are loaded into them. For "large", read "1 MB or more". DP have made a few tests with such documents, and come to the conclusion that not only is Perfection much faster with them but that some of the Windows programs cannot handle them at all.

My own experiences support this; get to 1 MB with a document in a Windows WP program and you either have to set your own internal clock to non-turbo or give up completely. These comments are made on the basis of tests with a **Gold Card** QL system and both 486DX2/66 and 486DX/40 PCs; the latter two systems have nominal clock rates of 33/66 and 40 MHz respectively, compared to the 16 MHz of the Gold Card.

The file which generally gets used for comparison purposes on my system is now close to 2.5 MB and 450 pages in size; while it is handled quite briskly by a DOS WP program, it brings the whole system almost to a halt when loaded into the equivalent Windows program. Indeed, there is no sense at all trying to work with it in that program. Anyone who has seen the DP demos of Perfection will be aware that the speed of navigation around documents such as a large chunk of the St. James Bible is quite remarkable. The program does not seem to mind what size the loaded file is. For the moderate sum of £9.95, anyone can now obtain some sample, compressed files and a demo version of Perfection; the supplied disk contains both **decompression and compression routines** (ARC and ZIP), about half of DP's database (in scrambled form), the Book of Genesis, the Maastricht Treaty, Startrek, and the Perfection demo files. The largest file is a mere 600 KB when uncompressed, because of the restraints of getting the files onto disk, but you can stitch three files together to get one of over 1 MB, for experimentation. ED disks are desirable!

Keep on Driving

When a floppy **disk drive fails**, there is no real need to go out and buy a complete new unit. "Bare" drives (without a power supply) can be obtained quite cheaply, and they are not that difficult to fit. It is now difficult to obtain DD (double density) drives, and HD (high density) drives are sold for less than DD ones used to cost. The last HD drive I bought cost £29, plus post and packing and VAT.

In many cases, **replacement** of the defective drive will be a simple screwdriver job. There can be electrical hitches, though. Old drive units, with built-in power supplies, usually had 5 volt and 12 volt supply lines, and some HD drives may need them both. However, current HD drives should need only the 5 volts. The presence of a 12 volt line should not create a problem, as that line ought not to be connected to anything within the drive.

There is an unfortunate lack of consistency in the way both power and signal connectors are located on drive circuit boards. Be very careful to check, and note down, the way these connectors are fitted on the existing drive, before removing it. With a bit of luck, there

will be markings on the circuit board of the new drive, to let you know where the vital terminations are. Look for a marking such as "5V" to show which way round the power connector should fit. On the signal connector, look for the digit "1" or "34" on the PCB; the edge of the connecting ribbon cable which is coloured red must go to the "1" end of the PCB connector.

Hard disk drive connectors look similar to floppy drive ones, but have a different numbers of pins - 40 - on current drives. However, the drives likely to be used with the QL are of an older type which should have the same 34-pin connectors as floppy drives. Make sure you have the correct cable before starting a job. When connecting a hard disc drive, the cable needs to have the 34- or 40-pin connectors at both ends. Where the 34-pin cable is used, there will also be another, smaller ribbon cable, whereas the 40-pin cable does not need a partner.

Non-mix dives

There are several types of hard disk drive and controller card, and they are not generally interchangeable. Do not be fooled into thinking an old drive or controller for a few pounds will be a bargain; possibly it will, but there is a good chance it will turn out to be the wrong type for your system. Drives advertised these days are almost all of the **IDE** (Integrated Drive Electronics) type, and they will not work with controllers fitted to most computers that are four or more years old. The older computers used FM (Frequency Modulation), MFM (Modified Frequency Modulation) and RLL (Run-Length Limited) controllers and drives, in that order, from oldest to least-old. The **SCSI** (Small Computer Systems Interface) type of controller and drive has been used on Apple computers for years, and was used by QL supplier CST, but is still relatively uncommon; there are now two interface types - SCSI I and SCSI II - on sale, and they are not totally compatible.

Atari/QL OS

Mention was made in a previous issue of what sounded like a new supplier, offering the **SMS2 operating system** on rom chip. The supplier's details are given in the **Information** box below.

I first heard opn the grapevine that a supplier called Furst (?) was offering the SMS2 operating system on rom chip for the Atari, at a price of about £135. There is a dedicated rom socket on some Atari motherboards, and the SMS2 chip fits into that; this approach sounds to be somewhat different from the QL emulators sold up to now.

What is being offered is only practically useful to users who have an **Atari** of some sort, but it should be of interest, even so. The operating system comes on perom (Programmable Erasable Read-Only Memory) mounted on a plug-in cartridge, to fit the Atari rom port. The programmable nature of the chip should allow future releases of the OS to be loaded from disk and then programmed onto the chip. It is stated that some storage space is likely to be available on the chip for the individual user's own system configuration details.

It may sound a bit too good to be true, but the suppliers state that all you have to do to get the OS working on the Atari is to plug the cartridge in - "... in less than 9 seconds you have a fully functioning, multitasking network-ready system". SMS2 will not run GEM or TOS applications, which were (are?) the mainstay of Ataris.

A few details sound very interesting. The OS supports pre-emptive multi-tasking. This, as I understand it, means that individual programs can get the processing time they require, rather than being allocated a fixed slice of the total time available. A much-simplified example may make this concept clearer: with three programs loaded, the "conventional" multi-tasking OS would probably allocate one-third of the basic time period to each, so that each program would be forced to take its slice of the time, even though it might be doing nothing.

Word processing programs are a good case of software that spends much of its time doing nothing; certainly, if a WP program is not the foreground job, it is likely to be idling, waiting for the user to type something in when next it becomes the foreground job. It might seem that spell-checking is a job that can be run in the background, but this is usually not true, since the checker will frequently be stopped and waiting for the user to press a key. When the WP program is in the background, it needs no processor time, but it gets it anyway, taking time away from the foreground job. **Pre-emptive multi-tasking** should ensure that the job which needs processing time gets it, and any jobs that are doing nothing get nothing. For those familiar with PCs, the obvious examples are Windows and OS/2, the former being an interface (not an OS) which has a rigid "fixed time-slice" method of allocating processor time to jobs, and the latter being a true, pre-emptive, multi-tasking OS.

The sharing of software resources is a concept that receives plenty of discussion. It is an attractive idea - provide essential functions, once only with the basic system, and let software developers utilise these in-built functions rather than "re-inventing the wheel" for every application.

Sharing

For instance, a **spellchecker** might be needed by your word-processing program, and also by a database and a spreadsheet; there seems little point in supplying three separate spellcheckers, but that is the way it has been done, on most micro computers. The SMS2 OS manages the sharing of such resources, both hardware and software. In theory, this should permit smaller software development teams to write only the main application program they are interested in, and simply provide "hooks" into ancillary functions, which may already be available from the system.

Odds and Ends

Bits of news trickle through on the **QXL** card. One of the people trying to sort out why a major group of programs would not run on the card apparently did not have much response to his queries about the operating system for some time. However, a later report suggests that **Turboed** programs have now been seen running. An irregular correspondent from north of the border has managed to run his 80286 PC at its higher speed of 10 MHz; previously, he had to run at 6 MHz, in order for the QXL to work, and the graphics performance at that level was not ideal. He is much happier now. Digital Precision's Perfection word-processor, and its associated utilities, have been cleared for use with the QXL card.

There are several programs available that can make use of graphics files within text, and most of us have been impressed by the enhanced appearance a few graphics can give to text. The problem is where to get the graphics files from. A few are supplied as samples

with programs, but they are often not good enough for what is required. Inevitably, envious eyes get cast on the vast library of such files available for other computers. The Belgian supplier, Progs, now offers conversion utilities for a limited number of file types, to go with their LineDesign program. For users not yet having this program (or wanting a wider conversion range than it provides), there is **Open World**, supplied by **Ergon Development** (see Information box). The program is stated to be able to convert **.GIF, .IFF, .TIF and .CUT** graphics files into QL monochrome, 4-, or 8-colour files. A disk of PC utilities is included, to convert QL screen dump files into .GIF files, and to "read QL disks on PCs, Sun, Vax..." Not many QL users will have access to machines of the power of Sun or Vax, presumably.

The word has been passed to me that my **chemical terminology** is out of date. Specifically, isopropyl alcohol is now normally referred to "in the trade" as **isopropanol**. Likewise with ethanol, methanol etc. If you happen to be a drag-racing fan, methanol should have been a familiar term for many years, but literature on disk cleaners doesn't often use the term isopropanol. Now we know.

Alcohol

The liquid itself is an evaporating alcohol cleaner, used in disk drive kits to clean the recording heads. It can be used for many other cleaning jobs, such as on the record heads of microdrives. It appears not to be strong, in the sense it is not liable to damage surfaces it comes into contact with, making it fairly safe to use for cleaning plastics.

When a floppy disk drive fails, there is no real need to go out and buy a complete new unit. "Bare" drives can be obtained quite cheaply, and they are not that difficult to fit. It is now difficult to obtain DD (double density) drives, and HD (high density) drives cost less than DD ones used to. The last HD drive I bought cost £29, plus post and packing and VAT.

Screwdriver

In many cases, replacement of the defective drive will be a simple screwdriver job. There can be electrical hitches, though. Old drives, with built-in power supplies, usually had 5 volt and 12 volt supply lines, and some HD drives may need them both. However, current HD drives should need only the 5 volts. The presence of a 12 volt line should not create a problem, as that line ought not to be connected to anything within the drive.

There is an unfortunate lack of consistency in the way both power and signal connectors are located on drive circuit boards. Be very careful to check, and note down, the way these connectors are fitted on the existing drive, before removing it. With a bit of luck, there will be markings on the circuit board of the new drive, to let you know where the vital terminations are. Look for a marking such as "5V" to show which way round the power connector should fit. On the signal connector, look for the digit "1" or "34" on the PCB; the edge of the connecting ribbon cable which is coloured red must go to the "1" end of the PCB connector.

Hard disk drive connectors look similar to floppy drive ones, but have 40 pins. Make sure you have the correct cable before starting a job. When connecting a hard disk drive, the cable needs to have the 40-pin connectors at both ends.

There are several types of hard drive and controller card, and they are not generally interchangeable. Do not be fooled into thinking an old drive or controller for a few pounds will be a bargain; possibly it will, but there is a good chance it will turn out to be the wrong type for your system. Drives advertised these days are almost all of the **IDE** (Integrated Drive Electronics) type, and they will not work with controllers fitted to most computers that are four or more years old. The older computers used FM (Frequency Modulation), MFM (Modified Frequency Modulation) and RLL (Run-Length Limited) controllers and drives, in that order, from oldest to most recent. The SCSI (Small Computer Systems Interface) (*see also DIYToolkit this month*) type of controller and drive has been used on Apple computers for years, and was used by QL supplier CST, but is still relatively uncommon; there are now two interface types - SCSI I and SCSI II - on sale, and, just to make thing no simpler, they are not totally compatible.

Information

Open World (graphics file converter), £18:
Ergon Development
(Davide Santachiara)
Via Emilio de Marchi 2
42100 Reggio Emilia
Italy.
Tel. (from UK) 010 39 522 70409

Publisher's Pack update, £10:
Software87
33 Savernake Road
London NW3 2JU.
Tel. 1400-1700 Monday-Friday only: 071 485 9008

SMS2 operating system for the Atari, £135;
Furst Ltd.
Delta House
Garfield Road
Bishop's Waltham
Southampton SO3 1AT.
Tel. 0489 894674
Fax 0489 895765

Perfection demo and sample large files, £9.95:
Digital Precision Ltd.
222 The Avenue
Chingford
London E4 9SE.
Tel. 081 527 5493

In Short ...

Do abbreviations drive you up the wall? Here are two you might have wondered about.

MIDI - Musical Instrument Digital Interface. A data protocol system used to connect electrical music synthesisers up for control by computers.

MIPS - Millions of Instructions Per Second. A measure of computing speed, not to be confused with Clock Rate. The latter defines the operating cycle of the CPU chips, but MIPS specifies how many actual instructions are performed per second. One instruction may take longer than one clock cycle.

TROUBLE!

HARDY HINTS

Perennial solutions to perennial problems. If you have any favourites, please send them in.

When large quantities of data need to be printed out, printing single columns can be very wasteful of paper unless, of course, you have so many fields in each record that a complete line is needed. The following 23-line programs will enable you to print out any amount of data as tightly packed as you want. The columns will read sequentially from top to bottom. Only three inputs are needed how many items, how many rows and how many columns. In your main program you may already have stored the value for the total records in some variable saved with the data. I store mine in one of the field zero elements. In this case, line 100 can be ignored and arranged for "items" to equal the total automatically. The printer instructions, 340-360, are based on the FX80.

For the purpose of the program example, I have assumed that records contain one field of a maximum nine characters with three spaces between columns. Condensed print allows 132 characters per line, elite 96 and pica 80. In condensed, therefore, we could have 11 columns which would take up (11x9) + (10x3) = 129 spaces. Default line spacing at one-sixth inches allows 60 rows, but at ten-72nds spacing 72 rows are possible. Before loading the data, allow an excess of 72 over the total records saved when dimensioning the array. This is to allow for the remote possibility where, in the example shown, you were just short of 71 for a final full page. The program would fill ten full length columns, with one single entry at the top of column eleven and with the remaining spaces being blank - supplied from the data array's \zero entries - otherwise an error message would be generated.

The program as printed would be a procedure within an existing program. If the data records that you require to print contain two or more fields, then lines 170 and 280 require one or more extra combinations of data and fill$. The extra fill$ items will need the appropriate integer to allow for maximum field length plus spacer. Lines 190 and 300 likewise will need expanding with one or more combinations of fill$ and data. Note that the latter extra are reversed because these lines must finish with a data printout to end the row.

Don Smith
Kirkbymoorside

Decimal	Hex	Keying	Display/Function	Decimal	Hex	Keying	Displ
128	80	CTRL ESC	a	192	C0	Left	Curs
129	81	CTRL SHIFT 1	à	193	C1	ALT Left	Curs
130	82	CTRL SHIFT	å	194	C2	CTRL Left	Dele
131	83	CTRL SHIFT 3	e	195	C3	CTRL ALT Left	Dele
132	84	CTRL SHIFT 4	ò	196	C4	SHIFT Left	Curs
133	85	CTRL SHIFT 5	ô	197	C5	SHIFT ALT Left	Pan

```
100  CLS#0: INPUT#0,'How many items?'to 25;items
110  INPUT#0,'How many rows per page?'to 25;row
120  INPUT#0,'How many columns per page?'to 25;col
130  a=1: b=a+row-1: pages=items DIV (row*col):
     left=items MOD (row*col): con: space10
140  FOR x=1 TO pages
150   FOR y=a TO b
160    FOR z=0 to col-2
170     PRINT#4,dat$(y+z*row);FILL$(' ',12-LEN(dat$(y+z*row)));
180    END FOR z
190    PRINT#4,dat$(y+(col-1)*row)
200   END FOR y
210   a=a+row*col: b=a+row-1
220  END FOR x
230  fullines=left DIV col: odds=left MOD col
240  IF odds: fullines=fullines+1
250  a=pages*row*col+1: b=a+fullines-1
260  FOR y=a TO b
270   FOR z=0 TO col-2
280    PRINT#4,dat$(y+z*fullines);FILL$(' ',12-LEN(dat$(y+z*fullines)));
290   END FOR z
300   PRINT#4,dat$(y+(col-1)*fullines)
310  END FOR y
320  IF fullines<row: formfeed
330  :
340  DEFine PROCedure con: PRINT#4,CHR$(27);'!';CHR$(4);: END DEFine
350  DEFine PROCedure space10:
     PRINT#4,CHR$(27);'A';CHR$(10);: END DEFine
360  DEF PROCedure formfeed: PRINT#4,CHR$(12);: END DEFine
```

793	802	811	820	829	838	847	856	865	874	883
794	803	812	821	830	839	848	857	866	875	
795	804	813	822	831	840	849	858	867	876	
796	805	814	823	832	841	850	859	868	877	
797	806	815	824	833	842	851	860	869	878	
798	807	816	825	834	843	852	861	870	879	
799	808	817	826	835	844	853	862	871	880	
800	809	818	827	836	845	854	863	872	881	
801	810	819	828	837	846	855	864	873	882	

The NEW USER GUIDE

Concepts Section

Section Thirty Three

This month the Concepts Guide begins QDOS

BINARY, HEX and DECIMAL

The relevance of the **binary counting** system to computers is well known. Ones and noughts can be represented by the presence and absence of an electrical current. Binary digits can be combined in groups of eight to represent values up to 255, groups of sixteen for values up to 65535, or groups of thirty-two for values up to 4,294,967,295. Binary values are also very good at representing **Boolean logic** - on the QL, non-zero values are interpreted as "yes", and zero values are deemed to be "no". See the next section for more details.

Unfortunately, decimal and binary are not easy to translate between. They share very few significant values in common. **Binary numbers** are mind-numbingly long series of ones and noughts. Programmers needed to find a way of representing binary values succinctly, so after flirting for a few years with base eight, or octal, they have almost universally settled for base sixteen, or **hexadecimal** (hex) as their counting base of choice. Hex values need a quarter of the number of digits to represent an equivalent binary number, and translations between hex and binary are very straightforward: every four binary digits become a single hex digit.

Talking of digits, mankind thoughtlessly stopped inventing numerical digits when they reached nine. The hex counting system needs fifteen digits in addition to zero, so the first six letters of the alphabet succeed 9. Some key decimal, binary and hexadecimal values are shown in **Figure one.**

Figure 1: Some key values in decimal, hex and binary

Decimal	Hex	Binary
Eight-bit values		
7	07	00000111
15	0F	00001111
16	10	00010000
32	20	00100000
127	7F	01111111
128	80	10000000
255	FF	11111111
Sixteen-bit values		
32767	7FFF	0111111111111111
32768	8000	1000000000000000
65535	FFFF	1111111111111111

Super Toolkit II contains functions to translate between binary, decimal and hexadecimal values. Alternatively, SuperBasic functions can be created that perform translations. The calculator project published in *QL World* a couple of years ago provided a full set of conversion functions. **Figure two** lists functions that convert between binary values, held as strings, to their decimal equivalents.

Figure 2a: Function to convert a string of binary values to its decimal equivalent.

```
100 DEFine FuNction Bin2Dec (binstring$)
110  LOCal result, bit, bitcode
120  result = 0
130  FOR bit = 1 TO 8
140   IF bit > LEN(bitstring$) : EXIT bit
150   IF bitstring$(bit) = "1"
160     result = result + 2^(8-bit)
170   ENDIF
180  END FOR bit
290  RETURN result
200 END DEFine Bin2Dec
```

Figure 2b: Function to convert a decimal value between 0 and 255 into its equivalent binary value

```
300 DEFine FuNction Dec2Bin$ (num)
310  LOCal result$, x
320  result$ = FILL$("0", 8)
330  FOR x = 0 TO 7
340   IF 2^x && num: result$(8-x) = "1"
350  END FOR x
360  RETurn result$
370 END DEFine Dec2Bin$
```

Boolean Algebra

The English mathematician **George Boole** (1815 - 1864) wrote two important treatises: *The Mathematical Analysis of Logic* (1847) and *An Investigation of the Laws of Thought* (1854). He applied his mathematical knowledge to what had previously been the philosophers' preserve of human logic. His name is now synonymous with computer logic, which tends to see things in black and white terms.

His logic vocabulary is not extensive: TRUE, FALSE, AND, OR, XOR and NOT form the bulk of it. True and false

are conditions easily represented by one and nought. The other words are operators similar to plus, minus and divide. The essentials of Boole's logical algebra are that statements that are true or false can be combined with Boolean operators to form further expressions that can be evaluated to be true or false. The following examples use **pseudo-code** (a form of note-taking that mixes SuperBasic keywords with standard English) to demonstrate the IF statement's debt to Boolean algebra:

IF it is cold THEN I will wear a coat: REMark what, even indoors?

IF it is cold AND I am outside THEN I will wear a coat: REMark but what if it is mild and raining?

IF (it is cold OR it is raining) AND I am outside THEN I will wear a coat

In short, if two statements that are true are linked with AND then the result is also true. Other combinations would involve at least one false statement and the result is false. If the link is OR, then if either statement is true then the result is also true. If both statements are false the result will be false.

In the final example, brackets are used to show precedence. There are actually two Boolean expression pairs even though there are only three things to test: the first pair is contained in the parentheses and the second pair is the result of evaluating the bracketed expression compared with the third statement.

The pseudo-code contains implicit true/false conditions that are better brought out by rewriting the IF statements along the following lines:

"IF cold = TRUE AND outside = TRUE THEN wear_coat = TRUE".

The only **changes** now needed to form a valid SuperBasic line are to assign values to the variables cold, outside and wear_coat and replace TRUE with 1. In SuperBasic, any zero value can be interpreted as being false and any non-zero value as being true.

Turning to the real world of computer data, most people's linguistic instincts mislead them into using AND when they mean OR and vice versa. Imagine a customer database being searched for records that fulfil two conditions. In English we might say "Show me all customers in Yorkshire and all customers in Lancashire", but Boolean logic insists that the AND is replaced by OR: "Retrieve all records where Location = Yorkshire OR Location = Lancashire".

Computers also have to represent a logical condition that is awkward to express in English: where one thing is true or another thing is true, but not both together. The concept has been given the name XOR, or Exclusive OR. The table at **Figure three** shows truth tables that demonstrate more clearly the workings of the logical operators.

Figure three: Truth tables for AND, XOR and OR

AND	XOR	OR
101	101	101
100 &&	100 ^^	100 ‖
100	001	101

Note: SuperBasic bitwise operators are shown: these can be used in programs or direct commands

There is no sense of "carrying" values between columns in bitwise logical expressions.

QDOS makes most obvious use of Boolean logic in its handling of screen colours (See the section on Colour for details) when the OVER -1 command is invoked. This XORs the current screen colours with whatever is on the screen to produce a new combination of colour.

BREAK, SWITCH AND PAUSE

The following key combinations are trapped by QDOS for special purposes.

A SuperBasic program can be interrupted using the Ctrl-Space keys. On some external keyboards this facility is also mapped to a dedicated Break key. Sadly, Sinclair have given no such power to the Esc key, even though it was the most obvious candidate. Their preference was to allow software developers to trap the Esc key and attach whatever significance they wished to it.

The Ctrl-F5 keys can be pressed to interrupt a window that is busily scrolling a column of information off the screen faster than you can read it. On external keyboards, this facility might also be mapped to a Pause key. *Super Toolkit II* implements this feature automatically for commands such as DIR and WLIST, prompting you to press a key to release each screenful.

If several jobs are running simultaneously or concurrently, Ctrl-C can be pressed to move between jobs. Some quite respected programs run simultaneously with SuperBasic. This gives you instant access to the command line and saves the programmers from writing basic file handling routines. The disadvantage is that the screen can get to look very untidy and confusing with SuperBasic commands mixed in with the application's text. The Psion suite, unless doctored to behave differently, only run in a stand-alone mode. Try launching *Quill* with the EXEC command rather than EXEC_W and then experiment with Ctrl-C.

Ctrl-Alt-F7 is the rather strange combination left in by QDOS developers at the end of debugging (if that process was ever formally ended). It has the unwelcome effect of crashing the QL.

DEVICES AND CHANNELS

A device is anything capable of receiving or sending information. A channel is the link between a device and the CPU along which information is sent or received. The revolutionary thing about QDOS was that vastly different devices could be treated more or less identically because as far as the computer was concerned it connected itself to channels: what lay at the other end became less important. This rationality contrasts sharply with the mess PC users have to contend with.

The class of "devices" is large and disparate. Obvious devices seem to include printers, monitors, and disk drives because they are distinct, physical things. However, only the printer meets QDOS's strict definition of a device (and pedants will tell you that it is the serial socket that the printer is connected to that is the true device!): the others are better termed peripherals. QDOS, for instance, has no way of communicating to the monitor screen without first opening a window: the screen might have dozens of windows open and each is a separate QDOS device. The same is true of a disk

drive or microdrive: it is not the storage medium that QDOS connects to but to a specific file on the medium.

Devices also include things that cannot be touched, such as network connections to distant computers or ram disks (areas of memory formatted to resemble microdrives). Interestingly, a special case has had to be made for the keyboard. It might be a peripheral but it is not quite a device in QDOS terms. Type in a command and the keyboard can be deemed a part of Window #0. If the instruction is an INPUT statement, then as soon as it is executed keyboard activity is reflected in Window #1. It is not the keyboard from which QDOS receives key presses, it is from the channel attached to Window #1. Should you then press Ctrl-C to switch to another task, the keyboard should move with you, otherwise you cannot communicate with the task. This behaviour has two effects worth knowing about. Firstly, if you close Window #0 in a program you might be able to re-open it as a console device, but until the QL is rebooted you will not be able to issue SuperBasic statements from it. Secondly, QDOS has two sorts of windows: consoles, which have two-way communications, and screens, which do not: if you declare a channel to be linked to a screen, you can send it text and graphics, but the INPUT and INKEY commands will be ignored.

Channels lie at the heart of the QL's flexibility with devices. Like an irrigation system, QDOS channels convey information between the CPU and devices. A channel can be opened to a window on the screen and text sent to it. The same channel can be diverted to a printer, or to the network, or to a file, and the same process repeated: programs shovel character codes into the channel and more or less wave them goodbye. QDOS invisibly and silently ensures that the information is presented to the device at the other end in the manner it best understands. One of its jobs is to filter out incompatible data or instructions, such as sending PAPER and INK commands to a printer or file.

There are one-way (simplex) channels such as those attached to screens and networks and two-way (half-duplex) channels such as those attached to consoles and serial ports. The QL does not permit simultaneous two-way communications (full duplex).

When the QL is first powered-up three **default channels** are created: #0 for command input, #1 for default screen output and #2 for program listings. If other channels are required, for instance to link to a printer or a file, they must be created and configured (or their defaults accepted). The syntax for doing this is complex and sometimes inconsistent. Firstly, we must review the OPEN family of commands:

OPEN #x, dev_name tries to open a two-way channel numbered #x to the device called dev_name. If the channel links to a screen or a network, only one-way communication will be permitted.

OPEN_IN #x, dev_name tries to open a one-way channel to an existing file on a microdrive, disk drive or ram drive, in effect making the file "read-only". This command is only for use with files.

OPEN_NEW #x, dev_name tries to create a new file and then link a two-way channel to it. This command is only for use with files.

The construction of a valid dev_name for the OPEN commands often fills programmers with dread. However, useful defaults are accepted by QDOS in the absence of some of the details. A console can be opened, for example, with the simple command OPEN #5, CON_ and then adjusted for size and location with the easier-to-handle WINDOW command. For those who want their devices properly specified, here are the commoner options:

Consoles:
con_wideXhighAxposXypos_buffer, eg con_200x100a50x50_64, where wide and high determine the size of the window, xpos and ypos determine its screen location and buffer indicates the size of the keyboard type-ahead buffer. The defaults are the equivalent of con_448x180a32x16_128.

Screens:
scr_wideXhighAxposXypos, eg scr_340x140a48x16, where wide and high determine the size of the window and xpos and ypos determine its screen location. The defaults are the equivalent of scr_448x180a32x16. Note that screens are one-way devices that ignore keyboard input and so do not have a type-ahead buffer.

Serial ports:
serNPHZ, eg ser1ehr, where N is 1 or 2 to indicate serial port 1 or 2, P reflects the parity protocol (the first character of Even, Odd, Mark and Space), H reflects the handshaking protocol (H for handshake and I for ignore) and Z reflects the data transfer protocol (R indicates no end-of-file code will be sent, Z indicates that a Ctrl-Z end-of-file marker can be expected, C indicates that QDOS's CHR$(10) newline character will be converted to the more usual CHR$(13) carriage return). All of these settings are optional and they can appear in any order. Note the lack of an underscore and a fixed parameter sequence compared with other device declarations.

Network connections:
netD_S, eg neti_0, where D indicates the direction information flows (I for input and O for output) and S is a station number (on a two-station network, each QL can be 0, but for larger nets each QL needs a unique station id less than 63). Note that this is an entirely different syntax for one-way data traffic to the OPEN_IN equivalent associated with files.

Files on microdrives, ram drives, and disk drives: devX_filename, eg FLP1_MyFile, where dev can be MDV, ram or FLP, X is a unique sequential number for that particular device type, and filename is a valid filename. Similar device names, ie three letters followed by a number and an underscore, have been associated with third-party disk drives and other devices.

CHARACTER SET

The QL's character set is based on the well-known **Ascii** (American Standard Code for Information Interchange) conventions, albeit with its own wrinkles added in for good measure. Incidentally, Ascii is on its way out in the computer world, being replaced with its close cousin the ANSI (American National Standards Institute) character set. Characters and bytes are closely related: there is a one-to-one relationship between them and they can readily represent each other. For instance, the QL's screen map comprises 32,768 bytes, and chunks of the screen can be saved to

a character string for manipulation. This feat is made much easier with *Turbo Toolkit's* PEEK$() and POKE$ routines. Single character/byte translations can be performed with the CHR$() and CODE() functions. Most people seem happy to refer loosely to one when they mean the other; disk capacities make more sense to the non-computer-literate if they are expressed in terms of the maximum number of characters they can hold.

Any 8-bit character set is limited to 256 characters, which is little problem as that equates to almost ten times the size of the alphabet. The first 32 characters in the Ascii set are "control codes", things that move the input cursor or delete characters, for example. The most famous is CHR$(27), the escape key. The next 32 characters include the digits 0 to 9 and many of the more common punctuation marks. The next set of 32 characters (from CHR$(65)) begins with the alphabet in capital letters and ends with more punctuation to act as a filler before the lower case alphabet begins the final 32-character chunk that ends with the copyright symbol, CHR$(127). The second half of the ASCII set, from CHR$(128) to CHR$(255), is less well regulated. Like many other sets, the QL's characters above CHR$(127) include letters with diacritical marks, printers' proof symbols, mathematical Greek, and so on. Unusually, the function keys get their own character values high in the Ascii table set.

The division of the lower table into 32-character chunks is no accident. Pressing Shift-A produces CHR$(65). Subtract 64 from that value and the Ascii code for Ctrl-A is found. Add 32 to the original value and the Ascii code for lower case A will appear. In bitwise terms, pressing an unshifted alphabet key produces a byte with the sixth and seventh bits set. Pressing the same key with Shift keeps the bit pattern the same except that the sixth bit is unset. Combining Ctrl with the original key unsets the sixth and seventh bits but keeps the lower bits the same. The lower Ascii table has the eighth bit unset and the higher table has it set. The bit pattern for the Ctrl-Shift-character series begins 101 but is otherwise identical to the main alphabet.

Knowing this series can be useful when, for instance, upper case text must be forced. Each character can be tested in a loop like this:

```
text$ = "A long piece of Text in Upper and Lower Case"
FOR x = 1 TO LEN(text$)
SELect ON text$(x) = 97 TO 122: text$(x) =
CHR$(CODE(text$(x))-64)
END FOR x
```

Ascii codes are only the first step in placing a character on the screen: the shape of the character is represented in binary form in the QL's rom, and other bit mapped character sets can be loaded into the QL's memory to temporarily replace the default set. Standard character shapes are made up of nine lines each of five pixels, represented in memory as nine bytes, the outer bits of each of which are ignored. The bitmap works by painting pixels represented by a 1 in the character set in the current INK colour. Zeroes in the bitmap are painted in the PAPER colour.

Working out where the character font is held in memory is something of a treasure-hunt through the QDOS tables. Each screen or console can be connected to a different font, so font addresses are repeated in each channel table. The location of each channel table is held in a master table, and the location of that is held in the main QDOS table at the start of what is known as the system variables area. While most circumstances will find system variables beginning immediately after the screen map this should not always be taken for granted, so the most reliable way of accessing them is through the Basic family of functions in the *Turbo Toolkit.* Alternatively, if you have a Minerva rom, you can obtain the base address of the systems variables with the function VER$(-2).

The first clue in the treasure hunt is usually located at address 163960, the start of the QDOS channel table and a pointer to the start of the channel table for Window#0. Calculate an offset four times the channel number to find where the channel table you want is to be found. Now find the long word offset 42 bytes from the start of the table: this is the start address of the first font used by that channel. Immediately after this pointer is another indicating the location of a second font: the QL divides the Ascii set into two fonts, initially at CHR$(128).

If the above paragraph left you lost, **Figure four** is a function that should find the start of the first font for any given screen channel.

```
400 DEFine FuNction FontBase (chan)
410 RETurn
PEEK_L(PEEK_L(PEEK_L(163960)+4*chan)+42)
420 END DEFine FontBase
```

Note that you will get a spurious result if the function is called with a reference to a channel attached to the printer, network or a file. The function assumes that the system variables are where they normally are in order to avoid specific reference to Turbo Toolkit or Minerva, neither of which may be available to you. If you favour either enhancement to the QL, adapt the function by including the appropriate keywords and offsets.

Should you wish to modify your computer's fonts, the first step is to copy the default font out of rom into ram where it can be edited. Then PEEK() and POKE values into the new font table to make the changes you require. Finally, reset the address in the appropriate channel table to point to the new font's location. This amounts to a non-trivial exercise, so an easy alternative is to refer back to the March 1990 edition of *Sinclair QL World* March 1990 to find a SuperBasic article on the subject complete with all the functions and procedure listings needed to doctor your fonts.

The simplest changes to characters can be made with the CSIZE command, although this has no effect on the shape of the characters, merely their proportions and spacing. Characters can be doubled in width and height and their horizontal spacing increased by an additional pixel or two. Third party utilities allow much more flexibility in font sizing and spacing: Turbo Toolkit and even SpeedScreen are two examples of what can be done.

33 Section Thirty Three

DIY TOOLKIT

Simon N Goodwin introduces SCSI to SuperBasic.

```
* DIY TOOLKIT fast low-level SCSI transfers for Qdos etc, V1.01
* Based by Simon N Goodwin on PD code from the TF Services BBS
* DO NOT use SCSI for any other purpose while running this code
*
start    lea.l    define,a1
         movea.w  $110.w,a2
         jmp      (a2)
*
busy     equ      3                  SCSI status busy bit
status   dc.l     $ffffff00          Thor XVI SCSI control register
datareg  dc.l     $FFFFFF08          SCSI port data register
drive    dc.b     1
result   dc.b     0
buffer   dc.l     0                  Used to store the SCSI address
*
bad_par  moveq    #-15,d0            Bad parameter error code
error    rts                         Return error code in D0
*
putscsi  moveq    #10,d5             SCSI WRITE opcode & flag
         bra.s    getints
*
getscsi  moveq    #8,d5              SCSI READ opcode & flag
getints  movea.w  $118.w,a2          Vector to get long integers
         jsr      (a2)
         bne.s    error
         subq.w   #3,d3              Check number of parameters
         bne.s    bad_par
         move.l   8(a1,a6.l),d3      Number of blocks
         beq      easy
         cmpi.l   #256,d3            Maximum is 256
         bhi.s    bad_par
         movea.l  0(a1,a6.l),a0      Get address - no checks!
         move.l   4(a1,a6.l),d2      First block to be used
         lea.l    buffer,a3
         move.l   d2,(a3)            Store length in buffer
         tst.b    (a3)
         bne.s    bad_par            Reject if beyond 24 bits
*
         trap     #0                 Need to be in supervisor mode
         or       #$700,sr           Disable interrupts
         move.l   status,a4          Point at SCSI status register
         movea.l  datareg,a5         Point at data register
wait_free
         btst     #busy,(a5)         Test busy bit
         beq.s    wait_free            until busy=1
         move.b   drive,d0           Fetch drive number
select
         move.b   d0,(a5)            Select drive
         btst     #busy,(a5)         Test busy
         bne.s    select               until busy=0
wait_cmd
         move.b   (a5),d1            Get status
         bmi.s    wait_cmd             until req=0
*
* D1 has status from drive; this should be 'command phase', 5
*
         subq.b   #5,d1
         bne.s    error1
*
```

SCSI stands for Small Systems Computer Interface. It's an American national standard, sweeping the micro world less than a decade after its inception.

SCSI is a fast, device-independent interface for computers and peripherals. It is most often used to link one computer to a hard disk drive, but it can do much more, supporting CD-ROM drives, graphics scanners, multiple computers and tape streamers for fast backups.

The device-independent nature of SCSI means interfaces read and write each device the same way, regardless of its type. Some devices have extra commands - for instance you might want to format a disk, but not a scanner or another computer! But they all support the commands that transfer blocks of data.

SCSI 2

The SCSI standard was published in 1986, aimed at mini-computers and work-stations. It was quickly taken up by CST for their Thor 1HD, Thor XVI and QL hard disk interface. Since then it has had little impact on the Qdos world, but this is set to change. SCSI cards are already available for the QXL and Amiga Qdos emulators, and Miracle Systems is poised to lauch a SCSI interface for the authentic QL later this year.

SCSI was first defined by the snappily named ANSI X3T9.2 committee. My copy of the specification, obtained from Chas Dillon, is numbered X3.131-198x, but it may have been updated since.

There are now two versions of SCSI; the new one, SCSI 2, can do all that the original could manage, and more besides. It runs faster: double speed by default, quadruple speed (known as SCSI 2 FAST) if you replace the usual resistor packs with

'active bus terminators'. It's faster still if you expand the cable to the SCSI 2 WIDE specification, which has yet to gain much support from hardware manufacturers.

Even the original SCSI can outrun early Winchester disk interfaces like ST-506 and IDE, but drives cost more because SCSI devices are 'intelligent' - each has its own processor and software. The price difference is shrinking as the limitations of older interfaces become apparent, and SCSI drives become more abundant.

8-bit Transfer

The DIY routines transfer eight bits at a time, to suit everything from the Thor's SCSI 1 to SCSI 2 FAST on the PC and Amiga. The SCSI wiring uses a 50-way 'bus' to connect up to eight devices at a time. The interface controls these wires for you, so all the low-level programmer sees is two extra hardware registers or 'ports' for control and data-transfer.

As any SCSI device is permitted to talk to any other, when the bus is free they must stake their claim by signalling on one of the eight data lines. Each device in a chain should have an unique number, usually set with DIP switches, in the range zero to seven.

The highest number gets priority, which is why manufacturers like Apple often give their computers device number 7. This simplifies things because they don't need to defer to others, but causes problems if you try to share one chain of peripherals between several computers. In the long run, Apple hope to do this by radio, but the FCC is unconvinced...

There are two commands in this project: PutSCSI and GetSCSI. They transfer 512 byte SCSI blocks to or from memory. Both take three numeric parameters: the address for data transfer, the first block number to use, and the number of blocks, up to 256.

GetSCSI 131072,20000,64 reads a 32K fractal screen from the middle of my Thor hard-drive in about 110 milliseconds.

The main difference between the commands is that PutSCSI uses the SCSI WRITE command, code ten, and GetSCSI uses command code eight, for READ. Other command codes do awesome things like copying blocks or searching for data patterns, without intervention from the controller till the operation is complete!

I'd like to add a general-purpose SCSI command-issuer and invite readers to let me know what they need. Some commands allow larger transfers, but the DIY commands use the simplest six-byte command format. This starts with the command code, followed by the logical block address in three bytes, where zero is the first block; then the transfer length in blocks from one to 255, or zero for 256 (128K), and a 'control' byte of zero at the end.

You need not worry about this as the code does it all for you, including encoding the 'logical unit number' of the target device into the top three bits of the block address. This

```
* We have command phase; send 6 bytes of command to data port
*
        move.b  d5,(a4)         Command type
        move.b  1(a3),(a4)      LUN/msb LBA
        move.b  2(a3),(a4)      mid LBA
        move.b  3(a3),(a4)      lsb of logical block
        move.b  d3,(a4)         Transfer length (0=256 blocks)
        move.b  #0,(a4)         Control byte
await_data
        move.b  (a5),d1         Get status
        bmi.s   await_data        until req=0
        btst    #busy,d1
        bne.s   error2          Not busy at data in phase
        lsl.w   #7,d3           Multiply count by 128
        subq.w  #1,d3           Adjust for DBRA down to -1
*
* Check whether we are reading or writing
*
        subq.w  #8,d5           Subtract command code for READ
        beq.s   reading           and see if it matches
        subq.b  #7,d1
        bne.s   error5          Data out phase not present
write_loop
        move.b  (a0)+,(a4)      Move bytes from memory to SCSI
        move.b  (a0)+,(a4)
        move.b  (a0)+,(a4)
        move.b  (a0)+,(a4)
        dbra    d3,write_loop   Send up to 128K at each try
        bra.s   wait_status
*
reading subq.b  #6,d1
        bne.s   error3          Data in phase not present
read_loop
        move.b  (a4),(a0)+      Get a byte and put it in memory
        move.b  (a4),(a0)+
        move.b  (a4),(a0)+
        move.b  (a4),(a0)+
        dbra    d3,read_loop    Read up to 128K in one burst
wait_status
        move.b  (a5),d1         get status
        bmi.s   wait_status     until req=0
        subq.b  #4,d1
        bne.s   error4          status phase error
        lea.l   result,a3
        move.b  (a4),(a3)+      get status byte
wait_message
        move.b  (a5),d1         get status
        bne.s   wait_message    until all=0
        move.b  (a4),(a3)+      get message byte
exit    andi.w  #$D8FF,sr       Enable interrupts
easy    moveq   #0,d0           BASIC sees no error
        rts
*
*  SCSI errors are recorded as negative status codes
*
error1  moveq   #-1,d0
        bra.s   exit2
error2  moveq   #-2,d0
        bra.s   exit2
```

```
error3   moveq     #-3,d0
         bra.s     exit2
error4   moveq     #-4,d0
         bra.s     exit2
error5   moveq     #-5,d0
exit2    lea.l     result,a3        Store result
         move.b    d0,(a3)
         bra.s     exit
*
define   dc.w      2                Two procedures
         dc.w      getscsi-*
         dc.b      7,'GetSCSI'
         dc.w      putscsi-*
         dc.b      7,'PutSCSI'
         dc.w      0
         dc.w      0                No functions
         dc.w      0
*
         end
```

```
DIY Toolkit SCSI-DIRECT, listing 2

100 REMark Sinclair QL World HEX LOADER v 3a
110 REMark by Marcus Jeffery & Simon N Goodwin
120 :
130 CLS: RESTORE : READ space: start=RESPR(space)
140 PRINT "Loading Hex..." : HEX_LOAD start
150 INPUT "Save to file...";f$
160 SBYTES f$,start,byte : STOP
170 :
180 DEFine FuNction DECIMAL(x)
190 RETurn CODE(h$(x))-48-7*(h$(x)>"9")
200 END DEFine DECIMAL
210 :
220 DEFine PROCedure HEX_LOAD(start)
230 byte = 0 : checksum = 0
240 REPeat load_hex_digits
250    READ h$
260    IF h$="*" : EXIT load_hex_digits
270    IF LEN(h$) MOD 2
280       PRINT"Odd number of hex digits in: ";h$
290       STOP
300    END IF
310    FOR b = 1 TO LEN(h$) STEP 2
320       hb = DECIMAL(b) : lb = DECIMAL(b+1)
330       IF hb<0 OR hb>15 OR lb<0 OR lb>15
340          PRINT"Illegal hex digit in: ";h$ : STOP
350       END IF
360       POKE start+byte,16*hb+lb
370       checksum = checksum + 16*hb + lb
380       byte = byte + 1
390    END FOR b
400 END REPeat load_hex_digits
410 READ check
420 IF check <> checksum
430   PRINT "Checksum incorrect. Recheck data.":STOP
440 END IF
450 PRINT "Checksum correct, data entered at: ";start
460 END DEFine HEX_LOAD
470 :
580 REMark Space requirements for the machine code
590 DATA 274
600 :
610 DATA "43FA00F434780110","4ED2FFFFFF00FFFF"
620 DATA "FF08010000000000","70F14E757A0A6002"
630 DATA "7A08347801184E92","66F0574366EA2631"
640 DATA "E808670000A40C83","0000010062DA2071"
650 DATA "E8002431E80447FA","FFCC26824A1366C8"
660 DATA "4E40007C0700287A","FFB22A7AFFB20815"
670 DATA "000367FA103AFFAC","1A800815000366F8"
680 DATA "12156BFC5B016664","188518AB000118AB"
690 DATA "000218AB00031883","18BC000012156BFC"
700 DATA "08010003664AEF4B","5343514567125F01"
710 DATA "664A189818981898","189851CBFFF66010"
720 DATA "5D01663010D410D4","10D410D451CBFFF6"
730 DATA "12156BFC59016620","47FAFF4916D41215"
740 DATA "66FC16D4027CD8FF","70004E7570FF600E"
750 DATA "70FE600A70FD6006","70FC600270FB47FA"
760 DATA "FF23168060DE0002","FF28074765745343"
770 DATA "5349FF1A07507574","5343534900000000"
780 DATA "0000","*",25823
```

leaves 21 bits, addressing up to 2 million blocks. Ten-byte commands let you blast past the terabyte barrier with 32 bit block numbers; for now, a gigabyte is enough to be going on with.

The code assumes that you have a working SCSI interface, and could get stuck if SCSI status is unavailable. The extensions bypass the operating system so you should not use them while the system accesses the drive. If you've just been using WIN1, wait for slave blocks to be updated before using these commands.

PutSCSI is as adept at scrambling a good hard disk as it is at repairing a broken one, so use it with care. When experimenting you must ensure that the blocks you use are not allocated by the filing system, or - better still - are allocated to a big file established for the purpose. Make a backup first, to be on the safe side.

Testing

I have carefully tested these DIY Toolkit extensions on my Thor XVI, which now runs a 20 megabyte Rodime hard disk thanks to patient help from Dave Moore and many readers. I have yet to corrupt the drive, but I've been careful. I intend to update the code as new SCSI controllers come to my attention; the interface is standard, but port or register addresses depend on the make and model.

As usual the code is listed in two forms - assembler source, and hex data for a SuperBasic loader. The Basic routine is the same for every DIY project (much like this sentence) so you should not need to re-enter it by now. The new stuff is the DATA from line 590 onwards. It's rather shorter than usual, at 274 bytes, which will come as a relief to those who earned their spurs re-typing the code for MORE or MultiBasic!

When you RUN the program it reads and checks the data. Wnen all is well it writes a file containing the code for the DIY SCSI extensions, which you can then link with Toolkit 2's LRESPR, LINKUP from DIY Toolkit, or commands like these:

X=RESPR(274)
LBYTES FLP1_SCSI_CODE,X
CALL X

Listing one is the source code, assembled and tested with Devpac 2. You should have no trouble re-assembling it on your own system, although other assemblers might insist on small changes, like an A after the MOVE in MOVE.L STATUS,A4 to placate Metacomco's fussy ASM. I sometimes forget these as better assemblers support Motorola 'generic' instructions which do not need to be forewarned about the address register.

Source code

The register addresses are encoded at the start of the file, rather than buried in the code, so you can patch them with the numbers that suit your system. Bytes 10 to 13 of the code hold the address of the SCSI status register, followed by the data register address in bytes 14 to 17, numbering from zero.

The next byte, nineteenth in the file, is the target device number. If you are not using a Thor XVI you probably need to change the defaults from 256, -248 and 1, by patching or re-assembling the code.

The code is not rommable and intended to run from ram, so lucky people with more than one external SCSI device can can POKE the nineteenth byte to redirect the commands, as they go along. It may also be useful to keep track of the addresses of the 20th and 21st bytes,

perhaps with SET from DIY Toolkit Volume U, as these hold the status and message bytes returned at the end of the SCSI transaction.

If something goes badly wrong the code stores a negative error flag between -1 and -5 (254 to 250 if you PEEK it) to show when it hit trouble. These flags correspond to the tests labelled ERROR1 to ERROR5 in the source. Other errors set bits in the 20th byte: zero means OK, eight signifies that the drive was still busy, two can stand for a variety of errors depending on the device, and 12 means 'reservation conflict' - the device is reserved by another SCSI controller.

Both SCSI registers are bi-directional. You move bytes of data to and from one, storing commands and reading status reports in the other. There is an elaborate software and hardware protocol to make sure that devices do not get out of step or interrupt each another.

First the code waits until the SCSI bus is not busy, testing bit three of the status port. Then it selects the drive by writing the drive number to the port, and waits again till SCSI is free and the REQ bit, the most significant status bit, is clear.

The loop labelled WAIT_CMD whizzes round till the sign of the status byte is positive. The BMI and BPL instructions let you test the sign bit of a byte without needing a BTST #7 instruction first. By now the device should be in 'command state', indicated by the value five in the status register, and we can send a command sequence.

The command accepted it only remains to move memory contents to or from the data port. Blocks of four instructions are used so that the DBRA does not dominate the execution time for the transfer loop. The extensions finish by storing status and 'message' bytes from the controller.

These are exceptionally low-level commands, and should never be needed if the WIN driver is adequate. Alas CST never implemented any equivalent of Tony Tebby's "*d2d" direct-sector access mechanism for floppies, so you're in trouble if your hard disk file structure gets damaged, perhaps by turning the machine off as it is writing. That's easily done on a Thor XVI, with its lazy slave-block updates.

The inspiration for this project came from Tony Firshman's QL Bulletin Board, where an anonymous Argonaut uploaded code to move individual hard-disk sectors to and from screen memory. That code was written in order to recover data from a hard disk with a faulty directory. Of course some devices, like tape streamers, have no directory to start with.

Another predictable advantage is speed. The DIY routines can go much faster than access through the WIN device, especially on large blocks of data. The drive keeps track of the location and there's no need to search for blocks as you go along.

I have used GetSCSI and PutSCSI to animate a sequence of screens at almost 300K per second, two to four times the speed of CST's WIN device. Much greater speeds are possible if you have a bigger drive, with more heads to read and write; my DKB 4091 interface for the Amiga can exchange up to seven megabytes a second with a big SCSI 2 FAST drive.

The third reason to use these commands is to access devices which are not yet known to Qdos, like CDs, tape streamers and scanners. While full device-specific Qdos (or Argos) drivers would be nice, it may be a while before those arrive, and meanwhile you can make simple use of them with GetSCSI and PutSCSI. It's quite practical to make backups by loading files into memory and streaming them out with PutSCSI. The DIY Toolkit INPUT$ function comes in handy if you need to split big files into sections, to save memory.

Next Month

SuperBasic in Action will return next month, with some more practical routines for lazy QL programmers. Meanwhile I'm eager to hear from SCSI pioneers, and plan to update these routines when I hear how you get on.

aRCHIVE aNSWERS

Robin Stevenson answers some basic Archive questions.

When producing elegant programming solutions to various Archive problems it is very easy to forget that a lot of readers are looking for a few simple pointers to help them get started in programming. So this time we shall have some real back to basics stuff, in direct response to some readers' questions. First we shall look at getting a yes-or-no decision from the keyboard. **Jusfiq Hadjar** wrote in, citing as his starting point a procedure, apparently dating from about 1985, in a booklet bundled with our forerunner, *QL User*. This procedure, YesNo, prints a prompt, waits for a "Y" or "N" key to be pressed, and sets the variable "yes" as 1 or 0 to denote yes or no. What Jusfiq wanted to do was extend the options, so that the Enter key also gave a "yes" answer, and the up and down arrow keys give a "no" answer.

One of the problems here is that these other keys don't return a symbol of any sort. So how do you test to see what key was pressed? The answer needs some explanation of how a computer stores letters and numbers in the character set.

Character Set

It does not recognise actual letters (capital A, little w and so on), nor does it bother about keyboard layout (2nd row, third key along) although in SuperBasic you can find that out. Instead all characters are given a number value, between 0 and 255. So the capital letters start at 65, going up to 90, and the lower case letters occupy numbers 97 to 122. The number keys and symbols fill in above, between and beyond these, making up the Ascii character set. Archive provides two functions to convert from the letter or symbol to its Ascii equivalent, and back. These functions are CODE() and CHR(). So CODE('A') will return 65, that being its Ascii code; and CHR(65) will return the letter 'A'. A full list of codes is in the Concepts section of the *QL User Guide*, as 'Character set'.

However the Ascii codes from 0 to 31 were designated 'control' characters, without having specific symbols assigned to them, all sorts of hardware and software companies drew up their own schemes for these characters. There are similar discrepancies in the codes above 127, but we've got the middle 96 reasonably standardised.

When Psion were devising Archive they drew up a list of the various control keys (edit keys, function keys, tab etc.) and gave many of them numbers in the Ascii range 0 to 31. So although they don't return a letter or symbol, they do have an Ascii code. So our first job is to find out which keypresses give which numbers. The first procedure in **Listing One**, 'TestKeys', does just that. For each key you press, the Ascii code is printed. If it is above code 31, the character it relates to is also shown. It is important not to try to print the control characters on the screen. Psion in their wisdom also used the same Ascii range to control various screen effects, via their 'screen driver'. So if you try printing control codes, you may get colours changing, extra lines being inserted, or even error messages.

Test the Keys

You can test combinations of edit keys, shifted or with control, to find the code of any input key. In our case, we can see that Enter is code 30, and Up and Down are codes 2 and 3. When you have finished, pressing 'Q' will quit the procedure. So we now know what we are looking for, in the Yes-No example. The next problem is how to extend the test from just looking for one letter, to looking for a selection. We could construct a long-winded 'if this OR that OR the other' clause for each of the tests. I have chosen to use another Archive function, called 'instr'. This unpronounceable function stands for 'in string', which is not much more informative. What it does is to search through one string of characters, to see if it contains another (usually smaller) such string. In our case the second string will be only one character, with the value of the key pressed.

So all we need to do is set up a string of the characters that count as YES, and another string of those that count as a NO, and keep reading the keyboard input, checking it with instr() until a YES or NO character is found. The full listing is shown in Listing One, with a short test procedure to show how it could be used.

The second question comes from **P D Stickley** of Weybridge, who wants to know 'How on earth can you make Archive print out hardcopy of data on A4 single pages?'. Now any question to do with printing is always a little fraught, because I can't be sure just where his problem lies. I am assuming here that the printer settings, cabling, and the Psion printer driver are all set up correctly, as he has obviously got Quill to print out his letter. Provided Archive has access to the same printer driver all should be well at that end of the operation.

Single Pages

That leaves two areas of concern. The first, and more basic is the whole question of getting Archive to send things to the printer. This is done almost exclusively through the 'LPRINT' command, and will almost always be made through an Archive program written specially for the job. Each report or list will need its own set of lprint instructions. Lprint can accept multiple text , string and numeric variables separated by semi-colons. Like PRINT, it will send a line-feed unless there is also a semi-colon at the end.

Sorting out report printing instructions is often long-winded, but is generally fairly straightforward if all you want is a single slab of text, where each line of the report will be a separate lprint statement. A standard letter would be an example of this. Where a list is required, you will normally have perhaps one lprint statement, which is repeated for each record in the database, from within a loop along the lines 'first : while not eof() ... lprint x$;y;z ... next : endwhile'. This general structure will probably be familiar to anyone who has done even a tiny bit of Archive programming.

So the other area of concern implied by the question is that of the single A4 sheets. This raises the question of left-hand margins, and page lengths. The

probable reason Psion avoided supporting such features directly is not because they thought it undesirable, but because it introduced a level of programming complexity they were reluctant to embark on. Instead they left the whole caboodle for the end user. Having said that, there are things that help, like the procedures in **Listing Two**.

Proc PrintInit should be called once at the start of a printing session. It should be tailored to include any specific printer set-up codes your printer needs. If your printer works how you want it whenever it is switched on, that line can be missed out altogether.

The more important function of PrinInit for our purpose here is to set up some global variables. They are 'global' because they are not tied to one procedure only (unlike 'local' variables, which only exist within one procedure). Within PrintInit you can specify how wide the margin should be, and how many lines you want to print before starting a new page.

Proc PrintOut

Having set up these variables, all printing should be done through Proc PrintOut. Each line of text you wish to print needs to be passed to PrintOut as a parameter. Provided none of your lines are too long for the printer width, it will automatically keep track of how many lines have been printed, and get Archive to request a new page after your specified number of lines has been printed. It does this by lprinting chr(12), another of those control characters, being put to yet another use by Psion. With your printer driver set for non-continuous paper, Archive automatically halts after each new-page instruction. If you have set your printer driver to continuous output, but wish to use cut sheets, it may be useful for the program to halt while you insert a new sheet. If this is required you should change the value given to 'PagePrompt' in proc PrintInit from 0 to 1. You will then be prompted to press Enter before the program continues printing.

The chief limitation with this system is that PrintOut can only have one entry per line, and will always move to a new line after use. It is not as bad as it sounds, as you can concatenate various bits of text and variables together, as shown in proc testprint.

You must also convert any numbers to text using one of Archive's number formatting functions (str, gen, dec, and int). Also it is up to the user to make sure the line is not too long. If it overspills and your printer continues on the next line, it will not only mess up the margin but will also over-run the lines per page.

If you are looking for a more complex procedure, that addresses the problem of line lengths, the Bibliography program on page 40 of Sept 1992 *QL World* contains proc PrintLine, which will do this.

Entering

Archive program procedures such as the two listings here cannot be entered directly from the command line. They must be typed into the Archive Editor. To use this, enter EDIT at the command line. You will then be asked for the name of the first procedure (proc).

After you have entered all the lines of the first procedure you can use the up or down arrow keys to select a particular line, and press F5 to make changes to the highlighted line. If you need to add more lines, pressing F4 will return you to insert mode. When you have entered 'proc TestKeys', pressing Esc will return you to the command line. You can try your procedure out by entering its name at the command line.

When you come to add the next procedure, by typing EDIT again, you will see proc TestKeys is still there. To create a new procedure you need to type F3 N, and you will again be asked for a procedure name. You need to do this for each of the procedures in the listing.

Each new procedure will be placed in alphabetical order in the list on the left hand side. You can move up and down this list by pressing tab or shift-tab.

When you have typed in or altered your program you must save it to microdrive or disk. To do this, choose an 8 letter name for the program file, and at the prompt type SAVE "filename". To load a previously saved program type LOAD "filename".

The program is divided into three main sections, spread over more than a dozen program files. Data handling comprises one section. Find/ Locate/ Searching, browsing, editing, inserting, indexing and selecting are all supported. As with all three sections, the program automatically produces customised procedures to handle your data, so that as you use it the program will tailor itself to meet your needs. File maintenance makes up the second section, allowing full or partial files to be created, copied, modified, transferred, and deleted, all from single key selection, and with files, fields, and options clearly shown on screen.

The third, and possibly most powerful section is for printed output. Whether it is address labels, single line or multiple line lists, spread over one or two columns, DBMAN can enable you to rapidly produce smart, formatted output, and can save all scripts to file for use (or even hand tweaking) at a later date. It also supports Archive's facility to redirect printer output to screen (for pre-print checking), or to file, for importing into Quill, or similar, for even greater output control.

Listing One - Enhanced YesNo Procedure

```
proc testkeys
  local Key$: while 1
    let Key$=getkey()
    if code(Key$)>31: print Key$;: endif
    print tab 10;code(Key$)
    if upper(Key$)="Q": return : endif
    endwhile
  endproc
```

```
proc YesNo;Prompt$
  rem **** Gets Y/[Enter] or N/[Up]/[Down] and sets "YES"
  local Yes$,No$,Key$
  let Yes$="Yy"+chr(30)
  let No$="Nn"+chr(2)+chr(3)
  print Prompt$;: while 1
    let Key$=getkey()
    if instr(Yes$,Key$): let YES=1: print " Yes": return : endif
    if instr(No$,Key$): let YES=0: print " No": return : endif
    endwhile
  endproc
```

```
proc test
  yesno;"Did you have a nice day? "
  if YES: print "Oh good": else : print "Oh dear": endif
  endproc
```

DBMAN, a Database Management System

On a completely different topic, **Hugh de Saram** of Marlborough has developed a staggeringly impressive 'front-end' for Archive. Its aim is to free the user from typing commands at the prompt, and making all sorts of options available by single key selections from menus. But this is far more than just a menu system for existing commands. Many tasks are completely automated, from the relatively mundane search for a particular record, up to creating a new database and then transferring selected data from an existing file, or printing out a database load of address labels.

Big Programs

Inevitably with a system of this size and complexity, there is an overhead. In fact there are several. One is the number and size of program files stored.

Even if you could get them all to fit on a microdrive, you would be waiting for ever while they swapped themselves in and out of memory. The second is the memory space required. These are big programs, so unless you have got some memory expansion it will fail to load.

Finally there is the sheer speed problem. Archive itself taxes the QL processor at times, and some of the goings on within these programs will tax your patience as a result. The complete answer to these problems is the Miracle Gold Card. With the disk drives, memory, (including slave blocks to speed disk access) and processing power this provides, DBMAN is in its element.

The PC version (to all intents identical) is being offered at £10, and would be good value at several times that price. Hugh has offered a copy of the QL version to people who send a disk and some postage. Quite frankly if you have a machine capable of running this, then go and find an envelope and a blank disk! Anyone prepared to read this far into an article about Archive would find this program useful. It is not so much that it makes Archive easier to use.

If you were left mystified by Archive, you may still be mystified by DBMAN. But if you were frustrated by the unwieldy interaction through the command line, this could be just what you need. And if you are an Archive enthusiast, I'm sure you'll love it.

DBMAN is available from Hugh de Saram, Littlefield, Bath Road, Marlborough, Wilts. SN8 1NN.

Hugh requests you enclose a 3.5-in disk, and postage on a self addressed envelope suitable for returning your disk. Anyone wanting the MS-DOS version should also include a £10 cheque. For the same price Hugh also has an even more sophisticated version for Arcplus - the souped up Archive - also under MS-DOS.

Listing Two - Sending output to the printer.

```
proc PrinterInit
  spooloff : let PagePrompt=0
  let MARGIN=10: let LineNumber=1: let PageLength=60
  lprint chr(0)+chr(27)+"x1"+chr(0)+chr(27)+"M"
  rem **** selects letter quality, 12 pitch on Epson 24 pin
  rem **** Include any setup commands your printer needs.
  rem **** remember to precede any control chars with chr(0).
  endproc
```

```
proc PrintOut;X$
  lprint tab MARGIN;X$
  let LineNumber=LineNumber+1
  if LineNumber=PageLength
    lprint chr(12): let LineNumber=1
    if PagePrompt: input "Press Enter to continue";X$: endif
    endif
  endproc
```

```
proc testprint
  PrinterInit
  local Count: let Count=1: while Count<86
    PrintOut;"This is line "+gen(Count,7)
    let Count=Count+1
    endwhile : lprint chr(12);
  endproc
```

Instant Access - The File Compressed!

HARDWARE

CL Systems 081 459 1351 Real Time Digitizer
Computer Technik (Jurgen Falkenburg) 010 49 7231 81058 (Germany) Hard disk interface and systems, tower housings.
Dilwyn Jones Computing (DJC) 0248 354023 Process controller, power regulator, network prover.
Miracle Systems 0904 423986. Gold Card; QXL PC card; disk adapter; Centronics adapter/lead.
Qubbesoft PD 0376 347852 Miracle Trump Card, Expanderam, 3.5in disk drives. Sales and support.
W N Richardson (EEC) 0753 888866. QL systems, monitors, keyboards and interfaces, disk drives and printers, peripherals.
TF Services 0344 890986 Hermes IPC, Minerva rom, keyboard membrane, repairs, spares.

SERVICES

Adman Services (Dennis Briggs) 0952 255895 Spares, repairs, support.
Joe Atkinson 36 Ranelagh Rd., London W5 5RJ. Roms, mdvs, spares, PD software.
Quanta: General Secretary: Joe Haftke 081 302 6154 User group, support, library.
Quo Vadis 0708 755759 QReview.

SOFTWARE

COWO Electronic 010 41 45 211478 (Switzerland) QTop, Atari QL emulator, Thor support
Deltasoft 7 Tyrell Way, Stoke Gifford, Bristol. FlightDeck, Image D, AMD Airplan
Digital Precision 081 527 5493. Perfection, PC Conqueror, Lightning, Professional Publisher and many others.
DJW Software 0256 881701 Homebanker
Dilwyn Jones Computing (DJC) 0248 354023 Discover, Textidy, QL-PC Fileserver, Fleet Tactical Command, Basic Reporter, QLiberator, Filemaster, The Gopher, The Painter, Flashback, DataDesign, QPAC2 and other Pointer Environment programs.
Di-Ren 081 291 3751 Fleet Tactical Command (Dist. by Dilwyn Jones)
Ergon Developments (Davide Santachiara) 010 39 342 492323 (Italy) ZM-X ZX Spectrum emulator, Open World, others.
Jochen Merz Software 010 49 203501274 (Germany) QL/Atari emulators, QSpread, File Finder, QPTR Pointer Environment Toolkit and other PE programs, QDesign 2, and others.
Lear Data Systems 6 Southview Green, Bentley, Ipswich, SUffolk IP9 2DR. PCB-CAD
Liberation Software 081 546 7795 QLib Basic compiler, utilities. (Dist. Dilwyn Jones.)
Ocean Computer Services 061 740 9002 Professional Poolster.
Pointer Products 0258 455117 Pointer Environment programs
Progs (Van Auwera) 010 32 16 48 8952 (Belgium) LineDesign, DataDesign and others. (Dist. by Dilwyn Jones; LineDesign dist. by Software87)
Qubbesoft PD 0376 347852 QL Home Finance, Public Domain software.
SJPD Software 0282 51854 Public Domain software
Software 87 33 Savernake Road, London NW3 2JU. Text87 Plus4, Publishers' Pack.

REMODELLING the QL

The Falkenburg HDD Interface and David Johnson's QL Box transform Rich Mellor's QL system.

INFORMATION
Product: QL hard disk Interface
Price: Varies
Supplier: Jurgen Falkenberg, Thanweg 36, D-7539 Ersingen, Germany
In the UK: W N Richardson & Co., 18-21 Misbourne House, Chiltern Hill, Chalfont St. Peter, SL9 9UE. Tel: 0753 888866

Product: QL Box
Price: Varies
Supplier: David Johnson, The Corner House, Loxley, Warwick CV35 9JT. Tel: 0789 842543.

When the QL was first released in 1984, home computers still tended to use cassette tapes to load and save. Although access speed varied, this was a very slow and unreliable method of storing programs, with even the shortest programs taking several minutes to load (or more often than not, to crash because the tape recorder was not at the right audio level). Although this was quicker than some methods used in the past (such as punched cards), improvements in speed and reliability were desperately required.

Disk drives were still relatively expensive, with 5.25in floppy disks as the industry standard. Sinclair opted for a new method which could enable the new QL) to be small, relatively quiet and have two built-in storage drives. The Microdrive stored programs and data in a small cartridge on an endless loop of tape. Although early Microdrives were plagued with problems, they were an improvement on cassette tapes.

By the time the QL was first shipped, the price of disk drives was dropping and the new 3.5in drives were fast becoming the new standard, much lighter, quicker and quieter than 5.25in drives. Disk drive systems began to be released for the QL in early 1985. They were still very expensive (one company was advertising dual 800K disk drives with interface for just under £700), and many QL users stuck with the (by now) fairly reliable Microdrives.

Within a short time, the cost of adding disk drives to the QL dropped dramatically. Now, most active QL users seem to have at least one 720K (1MB unformatted) 3.5in disk drive with their QL.

Floppy trends

720K 3.5in disk drives have long been the upgrade standard on the QL. Although IBM are now fitting 2.8MB disk drives as standard to their PCs, the QL market has seen a trend towards a new disk standard - the Extra Density (ED, with 3.2MB of memory) disk drive. These can only currently be used in conjunction with a Gold Card and are extremely quick - many users have actually expressed a preference for these over hard disks. The only problem is that until ED disks become standard on other computers, the drives and the disks are going to remain expensive. Indeed, I understand that Miracle Systems (who are the main supplier of these drives for the QL market) have had to withdraw the product from their range for now as they cannot obtain the drives at a price which they feel is acceptable to their customers.

The next step up from floppy disk drives is a hard disk. These are fast and store far more information than 720K 3.5in disks. The disk is fixed in the drive, unlike floppies (and 'removable hard disk drives' which are not currently supported on the QL).

The PC market has always had a wide range of hard disks (most software written for PCs is so inefficient at memory management or so large that it needs the capacity of a hard disk to run). The problem was designing an interface to connect QL to these hard disks relatively cheaply and easily.

There have been several hard disk interfaces and units on the QL market, including products from Quest, Miracle, ABC and Rebel. The Quest unit was the first for the QL, advertised as a 7.5MB hard disk for around £1150. I have never seen one of these units on the QL market, though.

Old interfaces

The ABC unit seems to have been similar to the Jurgen Falkenberg interface, but is no longer available. The Rebel unit was supposedly the quickest of the bunch, with a built-in ST506/412 controller board (needed by most PC drives). This did not make the market in any quantity. The Miracle hard disk seemed to be the simplest solution, and worked with the Gold Card and Trump Card. This unit plugged into the QL's rom port (making it slower than other interfaces), and included a 40MB hard disk and power supply, in a box about two-thirds the size of a QL.

The prices of these drives varied greatly, but none seems to be available any longer. The Miracle unit was last advertised for around £449 for a complete 40MB system.

It would be pointless to upgrade from microdrives direct to a hard disk, as programs and data must still be backed up. Hard disks are open to corruption like any storage medium. For 40MB, you would need around 400 microcassettes to back up a full hard disk.

Many users settle with a twin-drive QL, which, with a Gold Card and ram disks, provides fast and efficient storage. Once they have a couple of 3.2MB disk drives, many users find it difficult to fill these diskettes and the only programs which really benefit from the extra speed are the Psion programs which can access files stored partly on disk.

However, there is definitely a need for the speed and capacity of hard disks, and with the ever increasing range of QL software which can work with files part-stored on disk, the release of a new hard disk interface or any system parts is welcome.

Two types

To ensure that the interface meets with the needs of the greatest number of people in the QL market, Jurgen Falkenberg has produced a product which can be linked to two types of hard disks and controllers (MFM and RLL - see below) and will work with all current QL hardware. As with other types of hard disk interface for the QL, it is impractical to use Jurgen Falkenberg's interface with only 128K memory, in view of the amount of memory which the interface takes from the QL. However, there must be very few QL users who have disk drives but no additional memory, so this cannot be a severe restriction.

There are several different types of hard disk units available on the market, varying in speed, size and controller board required. The main controllers used on the PC are one of four types: MFM, RLL, ESDI, and IDE - the latter two are relatively expensive (having only recently been introduced). There should be plenty of MFM

drives available, as many PC users are now changing to the IDE type, which supports much higher capacities. RLL is however faster than MFM and may be preferred. The SCSI interface is only just starting to receive acceptance on the PC, as SCSI interfaces must plug several interface cards into the PC (this type of drive has long been the standard on Apple Macintoshes). CST produced a SCSI interface for the QL in the past, although some versions reportedly only supported drives with a maximum of 20MB. Miracle are currently developing one, but it will apprently wait until the QXL card and graphics interface are finished.

The hard disk interface produced by Jurgen Falkenberg allows you to link most MFM or RLL type hard disk controllers to the QL. Currently the MFM controllers supported are OMTI 5520A and 5520B, Western Digital WD-1002A-WX1 and Datatech's MFM-Controller. Similarly the RLL controllers supported are OMTI 5527A and 5527B and Western Digital WD-1002-27X and WD-1002A-27X. With a RLL controller linked to Jurgen's hard disk interface, the QL can access hard disks with capacities up to 416MB. There is no real need to link massive hard disks to the QL, as Qdos is built-in and does not need drive-room to operate (this is the main drawback with PC operating systems). However, for large storage capacities, I understand Jurgen can supply hard disks up to 122MB. For the real power-hungry, he hopes that soon the interface will allow two hard disk drives to be connected.

There is a wide range of set-ups, so it is essential to speak to either Bill Richardson or Jurgen Falkenberg before buying. The price, speed and capacity of hard disk drives vary greatly. Some of the newer RLL drives can reportedly be used with a Gold Card QL and the Jurgen Falkenberg interface to load data at up to 500K per second (about twice the speed of a ramdisk).

Right Angles

The hard disk interface itself connects to the QL's expansion slot. Due to the shape of the interface board, this will not plug in directly. If you have an existing interface plugged into the QL which has a through connector (such as the Expanderam), the hard disk interface should plug into the through connector (depending on the type of through connector, as the interface sticks out about half an inch from the side of the connector). If you have any other type of interface plugged into the QL, you will need a bus expander, a separate board which allows several interfaces to be plugged into the expansion port.

Jurgen can supply a 2-way bus expander (two interfaces into the QL) or a 5-way bus driver. The latter will allow up to five interfaces plugged, but is not Gold Card-compatible, unlike the 2-way expander.

Having decided whether you need a bus expander or a bus driver, you must decide how your QL will be set up. If you want to keep the QL's original case, you will need a bus expander/bus driver with a flexible cable. However, if you are rebuilding your QL, both vertical and horizontal bus expanders and connectors are available (the vertical ones are better suited to smaller cases). Jurgen provides a diagram of the different types.

The hard disk controller must then be plugged into the hard disk interface. This is normally fitted at right angles to the interface board, however, this can take a lot of room and cannot be used with the vertical bus-expander or the bus-expander with flexible cable. To reduce the room needed, an angle adapter is available which allows the controller to sit just above the hard disk interface. With this angle adaptor, the interface will fit any combination of bus-expander and interfaces.

Once you have connected all this, you will need to connect the hard disk drive to the controller by two flexible cables. This needs careful thought about where the hard disk goes, as the flexible cables have a maximum length (around one foot). The other problem, if you are not purchasing a complete hard disk system, is the power supply. Although the hard disk interface, bus-expander and Gold Card (for example) can be powered by the QL's own power supply, a separate power supply is needed for the hard disk drives. The type varies depending on the hard disk drive, but generally if a standard transformer is used, this must be of a very good quality.

Now comes the test.

Troubleshooting

If the QL does not work properly when switched on, provided that you have followed the instructions on putting the equipment together, you will probably find that this is due to one of two things. If you have a Gold Card attached, you may need to remove the 68008 (the large chip on the far left of the QL board) to reduce the load on the QL's power lines. This chip becomes redundant once the Gold Card is plugged in. If in doubt, talk to Miracle about it.

The other problem may be incompatibility with other interfaces plugged into the QL. However, Jurgen has anticipated this with a series of DIP switches on the hard disk interface which enable you to set the addresses in memory which are to be occupied by the interface. The manual gives a short line of SuperBasic which to test which slots in memory are free for use by the interface, and a table to help you to change the address.

Once the hardware is correctly installed, you will see a copyright message at the top of the start-up screen, which confirms that everything is working. The next step to take is to format the hard disk for use with the QL.

Informing the Interface

Before the SuperBasic FORMAT command can be used, you will first of all need to use the new WIN_FORMAT command to tell the interface about various physical characteristics of the hard disk drive attached to the controller. WIN_FORMAT requires a total of seven parameters which are described in the manual. These are used to tell the interface about the interleave factor, cluster size, sectors per track, number of heads and cylinders, and the delay to be used for start up and writing to the disk. There is no real need to worry about these parameters, as the relevant details should be provided with your hard disk drive, or if not, Bill Richardson (and Jurgen) should be able to provide a list if you tell them which hard disk drive you have.

Having used WIN_FORMAT, the FORMAT command will be needed to prepare the hard disk for use by the QL. This can take quite a long time (about one minute per 150 tracks on a drive with four heads), but luckily should not be required very often.

Once the hard disk has been properly formatted, its size is shown on the start-up screen as xxMB. I was pleased to note that on my system, formatting a 40MB drive actually provided me with 41MB for use.

Some older drives may take a while to reach operating speed when the QL is switched on. In this case, try resetting the QL once from the start-up screen to get it to recognise the drive (if the hard disk copyright message gives the drive size on the start-up screen, this is not necessary). Once installed, the hard disk can be used like a normal QL disk drive, using the device name win1_, for example:

```
LOAD win1_Program_bas
```

Other commands which are specifically for use with the hard disk interface allow you to set the stepping rate (which may vary with the age and type of the drive), clear out all the QL's slave blocks and write them to the hard disk, and alter the security level of the disk driver. Slave blocks are an area of memory used to store files currently in use. Normally, when you close a file, slave blocks are flushed to the hard disk to ensure that the files on the hard disk are up to date. However, the user can use the command WIN_SEC 0 to prevent this (speeding up disk access slightly), but will then need to use the command WIN_FLUSH to tidy up the slave blocks.

One of the problems with hard disk drives is the closeness of the read/write head to the disk. If you move the hard drive too much, this can knock the head onto the surface of the disk and damage either the disk or the head fatally. Hard disk systems generally have a command such as WIN_PARK to lock the drive heads into position, and Jurgen's interface is no exception. However, it is nice to see that Jurgen has

allowed you to pass a parameter to the interface, such as WIN_PARK 300, which will lock the heads if the drive has not been accessed for a specified time.

WIN_USE lets you change the name of the device used to access the hard disk drive to any other three letters (eg mdv). This makes it much easier to adapt old software to run from the hard disk.

Level-2

One of the main problems with hard disks is the very fact that they can store so much data. It can be difficult to find one file amongst a list of several hundred in the DIR. Luckily, Level-2 drivers allow you to create sub-directories using MAKE_DIR. For a discussion of Level-2 drivers see QL World January 1992.

Level-2 drivers are fast becoming a standard on the QL, being incorporated into the ST/QL emulator, Gold Cards, and replacement roms for the Trump Card and SuperQBoard. For users without access to these, the hard disk interface contains Level-2 drivers needed for the hard disk. You can create sub-directories, and set the backup date and version numbers of files (Toolkit II already allows you to set the update date on a file). The manual for the hard disk interface only has a brief note on these commands, and it seems Jurgen has assumed users will have access to Toolkit II (which comes with most disk interfaces) and Level-2 driver documentation.

One area which will confuse new users is the way the QL starts up. Once you have pressed F1/F2 from the QL's start-up screen, the Interface links a new device called 'boot' into the QL, which is found by the QL on start-up. This device then proceeds to LRUN win1_boot (if there is a formatted hard disk drive in the system). Due to the way floppy drivers work, the first access to the disk drive has to be made via mdv instead of flp. Some users may think the system is refusing to recognise the floppy drives. The interface manual says the simplest solution is to keep a file called 'boot' on the hard disk, which contains at least the line FLP_USE flp, which will return everything to normal. (If you find that the floppy disk drives are not being recognised by the QL, always try a FLP_USE flp command before trying to access them.)

Unfortunately, Jurgen cannot get round this as it relates to the way floppy drivers are initialised. It is however surprising that Jurgen has implemented the 'boot' device, which has been abandoned on the ST/QL emulator. This will not usually be apparent to users unless they try something like VIEW boot, which will fail to find the file boot on the default drive provided by Toolkit II.

Assistance

In all, the QL hard disk interface produced by Jurgen Falkenberg fills a niche in the QL market. The manual itself has suffered a little in the translation from the German, and I hope this will be improved in time. The hard disk interface would appear to be compatible with everything thrown at it (some users have reported problems with the QL recognising disk drives on Gold Cards with a red PCB, although I have one of these Gold Cards and have not experienced any problems).

With the options available, the non-technical may need a little help setting up the hard disk, although, from my experience, both Jurgen and Bill are more than willing to help with queries. This method of attaching a hard disk to the QL can be very cost effective; for example, I bought a bare 40MB 5.25in hard disk drive and controller, two-way bus-expander, angle adaptor and hard disk interface from Bill Richardson for the inclusive price of £301. This can be reduced even further you supply your own hard disk drive and controller board, although you may need to add the price of a power supply and box for the hard disk drive to this figure.

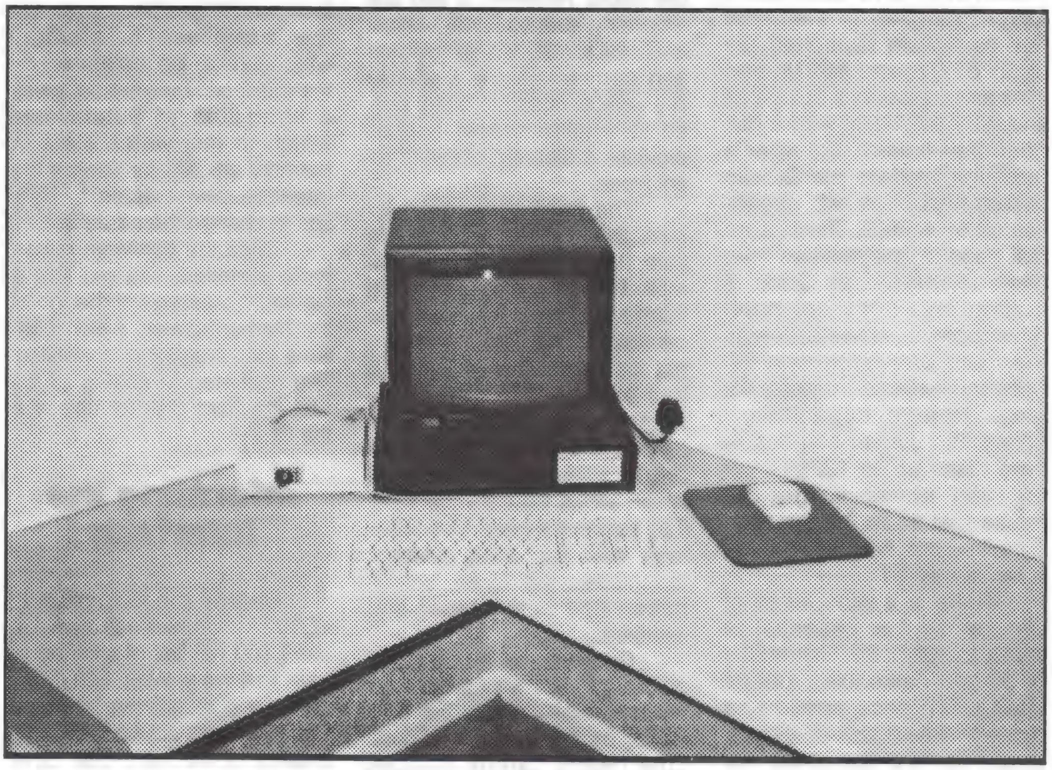
The QL Box

Having got my hard disk system, I realised that it was now slightly impractical (due to desk space) to keep all of my QL equipment in its original case. I already had an external keyboard (using the Jurgen Falkenberg interface) and so having spoken to Bill, I contacted David Johnson who confirmed that he had some made to measure QL boxes which would enable me to fit the whole computer in one box.

The actual box currently supplied by David measures 16.5in (width) by 13in (depth) by 3.5in (height). You will need to add approx 1.5in to the depth for the cables which come out of the back of the box. You can fit any QL add-ons inside the box, such as expansion boards, QIMI mouse interfaces, two 3.5in disk drives, and one 3.5in hard disk drive.

If a 5.25in hard disk is attached (as in my case), this will need to sit on the top of the box in a standard 5.25in disk drive case (also supplied by David). I do understand from David that the next boxes he produces will be slightly larger so that you can fit one 5.25in disk drive (either a floppy disk drive or a hard disk) inside the box, together with two 3.5in disk drives, making the QL even more portable.

Fitting a QL into a box was never really a project for the non-technical amongst us and David will actually fit everything into the box for you. Once the QL board has been removed from its case and the microdrives removed (these will not fit inside the box), the QL board has to be modified so that it takes power from the switched mode power supply unit fitted in the box, which provides the power for the drives, interfaces and anything else which may be attached to your QL. Not only does this cut down on the number of power points required to run the system (and even the power consumption), but it provides a much smoother power signal which helps to prevent the QL from locking up.



Having modified the QL board, the various disk drives

need to be removed from their casings and screwed firmly into place, taking care not to break any of the wires. The connector from a keyboard interface is screwed onto either the back or front of the box (whichever you desire) so that a keyboard can be easily plugged or unplugged as required.

Bus Expander

The box is not long enough to take the QL board plus any expansion cards, and so one of Jurgen's vertical 2-way bus expanders (or a flip-over board) is required, which plugs into the QL's expansion port, allowing you to mount expansion cards above the QL board. If you allow David to fit this for you, he will attach supports to the case to hold the expansion cards in place.

Once everything is in place and all of the wires re-connected, an on/off switch (which can be mounted at the front or the rear of the case) needs to be wired up, along with a power-on light on the front panel. A further panel light shows when the hard disk drive (if installed) is being accessed - your original floppy disk drive fascia is used at the front of the case, allowing you to see their access lights. The case then needs to be screwed together and tested. All of the normal QL sockets (except for the power socket and the expansion connector) are available at the rear of the box, as is the reset button.

The only thing which users may miss is the microdrives. However, fewer people are now using microdrives and in view of the extra difficulties in setting them up, it seems reasonable not to provide for these in a QL box (you can always access microdrives via the network port).

More Portable

Once the QL has been placed in the box, the whole system becomes much more portable, needing only one power lead and less room. Users who have access to a PC may wonder where the fan is. However, in several hours of continuous use, I have yet to suffer from any overheating problems (or any lockups following David's alterations to the QL board). It appears that the space in the box allows sufficient air to circulate to ensure that everything runs cool.

The actual cost of putting your QL in a box will depend on the hardware which you have and the amount of work which you are willing to do yourself. You will however have to supply your QL to David to enable him to make the necessary alterations to the board, which could be a little offputting to some users (David does however attend most Quanta shows and so you could arrange to hand over your equipment there). You will need to supply at least the following items of hardware in full working order (on the assumption that David is to do all of the fitting):

1. Sinclair QL
2. A Keyboard interface
3. 3.5in disk drives
4. 2-way expansion board (or flip-over connector) for the disk interface
5. Disk interface

David can supply the case itself plus the power supply unit for £85 if anyone wants to try fitting their equipment into the box themselves, but this is probably too technical for the standard QL user. If on the other hand, David were to fit all of these into the box and modify your QL board to take power from the switched mode power supply he supplies, the cost would be as follows:-

1. QL Box £50
2. Switched Mode Power Supply £35
3. Labour and parts £80 (approx.)

Total (approx.) £165

This price will vary depending upon the additional hardware you supply and whether you do any of the work yourself. You will also need to add a sum for post and packing if you are to deal with David through the post. It is therefore best to telephone David to discuss this further if you are interested.

SUBSCRIBE TO SINCLAIR QL WORLD AT LAST YEARS PRICES AND SAVE ££££££'s

££££

UK
£23.40
(12 MONTHS)

EUROPE
£32.90
(12MONTHS)

REST OF WORLD
£40.90
(12 MONTHS)

PLEASE SEND YOUR SUBSCRIPTION FORM TO SINCLAIR QL WORLD, THE BLUE BARN, TEW LANE, WOOTTON, WOODSTOCK, OXON. OX71HA.
TEL:0993 811181
FAX:0993 811481

I WISH TO SUBSCRIBE TO SINCLAIR QL WORLD
WE HAVE HAD TO RAISE THE COVER PRICE OF QL WORLD BUT WILL BE HOLDING LAST YEARS SUBSCRIPTION PRICE FOR THE IMMEDIATE FUTURE ONLY.

Existing subscribers will not be affected.

PLEASE DEBIT MY CREDIT CARD ACCESS/VISA

CARD NUMBER..

EXPIRY DATE

PLEASE SEND ME A 12 MONTH SUBSCRIPTION TO SINCLAIR QL WORLD.

NAME..

ADDRESS..

....

TELEPHONE NUMBER..

SIGNED..

DATE..

STARTING WITH ISSUE..

MICRO ADS

SINCLAIR QL + twin floppy drives, 256k memory, separate keyboard, many games £125 0491 681821

QL JM EXPANDED, SUPER-Q board 512k, 102-key keyboard, NEC 3.5" drive, Philips monitor, Epson RX-80 printer, Psion V2.35, QRAM, QUIMI, Manual, DL World Mags, £200 ono, Tel:0952 676668 evening.

QL plus twin three and a half inch drives. 768K Trump Card. Philips RGB input monitor. Borther M1009 printer. Software inclusing Turbo compiler. Manuals. #120 or offers. SUssex 0243 531. Also 512K Expanderam.

JM/JS QLs, manuals, extra RAMs, twin floppy, two monitors, Canon 1080A, cables, Psion software, Turbo Quill Plus, FLight Simulator, Archivist, Lightning, Archive tutor, Chess etc. Books, magazines. #250 SAE for list. 021 430 7772.

QL unexpanded JM version, shinwa CPA 80Q printer, joystick, Software including Psion suite, Eye-Q and Games, 40 MDV's, Manuals etc, £150 Tel:0256 764006 evenings only

QL Minerva Rom, Trump Card 3-1/2" D/D,Miracle centronics (x2) Tandata modem,serial printer, 6 years QLWorld, Quanta mags, books, software, b/w television extra MDV. Sensible offers please. Tel: 0977 515428

COMPLETE OUTFIT QL 640k js. Twuin Cumana 5.25" disc drives. 14" colour monitor. Tandata modems. Manneshan MT8. Printer lightening S.E. Backup QL manuals etc. All VGWO offers Cliff 0934 812640.

CLASSIFIED

Please send coupon to: Advertising Department Sinclair/QL World, The Blue Barn, Wootton, Woodstock, Oxon. OX7 1HA. Please place this Classified Advertisement in the next issue of Sincair/QL world.

NAME ....

ADDRESS ....

....

DAYTIME TELEPHONE NO ....

I enclose a cheque/P.O. to the value of £17.65 (inc. VAT) for a lineage advertisement of no more than 30 words. Please make cheques payable to: Arcwind Ltd.
